International Journal of Networked and Distributed Computing
https://doi.org/10.1007/s44227-023-00019-4

RESEARCH ARTICLE

Modified Playfair for Text File Encryption and Meticulous Decryption with Arbitrary Fillers by Septenary Quadrate Pattern

N. Sugirtham[1] · R. Sherine Jenny[1] · B. Thiyaneswaran[2] · S. Kumarganesh[3] · C. Venkatesan[1] · K. Martin Sagayam[4] · Lam Dang[5] · Linh Dinh[6] · Hien Dang[7,8]

Received: 29 August 2023 / Accepted: 12 December 2023

Abstract
Cryptography secures data and serves to ensure the confidentiality of records. Playfair is a cryptographic symmetrical algorithm that encrypts statistics based on key costs. This secret is shared with an authorized person to retrieve data. In the conventional pattern, there is an area complexity and deficiency in letters, numbers, and special characters. This hassle has been overcome in previous studies by editing pattern dimensions. The fillers used throughout the enciphering were not eliminated during the retrieval process, which resulted in the indiscrimination of the retrieved statistics. The proposed method uses a separate quadrate pattern that strengthens the Playfair cipher and guarantees that the fillers are eliminated to ensure the authentic retrieval of records. The fillers indiscriminate and strengthen the set of rules in opposition to brute force and avalanche impact. The proposed algorithm was evaluated with a minimal change in the key, and was found to have an avalanche effect between 65% and 93.7%. The encrypted document is further encoded using the Lempel–Ziv–Markov chain algorithm (LZMA) to provide compressed second-level secured text with a compression ratio of 0.75 for a file size of 100 KB. The pattern was designed to subsidize the integrated characters found on the keyboard.

Keywords Avalanche effect · Brute force · LZMA · Pre-computation attack · Classical encryption

1 Introduction

The expansion of communication and technology in the current era has sparked demand for information security. Establishing secure communication and preventing data from unauthorized access cryptography is essential. Cryptography refers to a range of approaches derived from mathematical concepts that are aimed at concealing information. Cryptography provides numerous algorithms to protect data from digital threats. Cryptographic algorithms such as classical encryption and modern encryption techniques are often used. These algorithms encrypt the data at the transmission end and decrypt it at the reception end. By encrypting the data, the original text (plain text) is transformed into meaningless random text (ciphertext). These ciphertext are transmitted through a network received by the receiver. During decryption, ciphertext is transformed into a meaningful message. The data are encrypted and decrypted using a randomized set of bits called an encryption key. However, these algorithms differ in their level of efficiency, which is determined by factors such as the successful retrieval of records and strong resistance to cyber-attack.

✉ Hien Dang
hiendt@tlu.edu.vn

[1] Department of ECE, Dr. Mahalingam College of Engineering and Technology, Coimbatore, Tamil Nadu, India

[2] Department of ECE, Sona College of Technology, Salem, Tamil Nadu, India

[3] Department of ECE, Knowledge Institute of Technology, Salem, Tamil Nadu, India

[4] Department of ECE, Karunya Institute of Technology and Sciences, Coimbatore, Tamil Nadu, India

[5] Department of Computer Science, INSA Lyon, Villeurbanne, France

[6] Department of Information Systems, Suffolk University, Boston, MA, USA

[7] Department of Mathematics and Computer Science, Molloy University, Rockville Centre, NY, USA

[8] Faculty of Computer Science and Engineering, Thuyloi University, Hanoi, Vietnam

Published online: 01 February 2024

The encryption processes for all of these algorithms use key values. Cryptography can be segmented into symmetric and asymmetric methods. Symmetric key cryptography uses the same key for both enciphering and deciphering, whereas in asymmetric key cryptography, the different keys are public and private keys for encryption and decryption. The remainder of this paper is organized as follows. Section 2 presents the related work based on a literature survey. Section 3 describes the proposed algorithm and its process. Section 4 presents experimental results. Finally, a summary of the study is presented.

2 Literature Review

The playfair cipher is a well-known symmetric encryption technique invented by Charles Wheatstone in 1854 [1]. This algorithm enhances the security of the transmitted information by encrypting letters in pairs. The conventional Playfair algorithm is composed of a 5 × 5 key matrix that can encrypt a maximum of 26 letters, with I and J placed in the same matrix square. This is a substitution cipher that encrypts pairs of letters, instead of a single letter. The Playfair cipher uses a 5 × 5 table containing a keyword, and the remaining spaces are filled with letters of the alphabet in order of I/J in the same location. To perform encryption, the plain text is broken into bigrams and substituted with an equivalent ciphertext from the key table. Because the plaintext is broken into a diagraph, an odd number of letters in the message require a filler. The choice of filler can be infrequent. A weakness of the traditional playfair algorithm is that it consists of only 26 uppercase letters. Standard filler of ‘X’ is used. Not all other numbers, symbols, or special characters can be included in the matrix. In general, I and J occupy the same space in the matrix; hence, one letter must be omitted from the reconstruction [2].

There are a few reasons for the non-preference of traditional play fair ciphers, as it can be easily cracked if there is adequate text and frequency analysis of the bigrams is possible. Binary 4 × 4 Playfair [3] proposed increases the run time with an increase in the input size. The proposed algorithm can encrypt letters, numbers, symbols, and any type of media files. The rotation of the key matrix creates randomness in the algorithm, which enhances the security.

The extended Playfair approach fails to encrypt special characters but has the advantage of not using I and J in the same block [4]. The 6 × 6 payload cipher proposed by considering numbers, space, and alphanumeric characters cannot encrypt special characters [5, 6]. Furthermore, the authors did not consider the avalanche effect in their study [7]. The authors utilized an 8 × 8 matrix with the concept of graph labeling to safeguard the data against unauthorized access. An enhanced square-key matrix with LFSR showed an improvement in the avalanche effect with a matrix size of 10 × 10 [8].

The authors used a 10 × 10 matrix with all possible keys exhibiting a minimum avalanche effect [9]. All possible characters on the keyboard that allow the repetition of characters in the keywords are considered. A modified 16 × 16 matrix [10] that considers all possible numbers, characters, and letters was proposed. The main limitation is the size of the matrix. They used symmetric encryption with a single key. In addition to the above limitations, Playfair is prone to frequency-analysis attacks [11]. Hence, to overcome these limitations and mention the Playfair cipher in relation to data authentication and copyright protection in 5G networks [12] [13], its applications in wireless sensor networks [14], its use in homomorphic encryption [15], and steganography [16, 17] have sparked our interest, prompting us to propose a new modified cipher algorithm following the rules of Playfair. Although Playfair is a classical encryption technique, it demonstrates a better performance when addressing time complexity [18]. In addition, the Playfair cipher offers effective file-text security measures [19, 20]. The main limitation of [19] is that the processing times for encryption and decryption are linearly proportional.

Data Communication, processing, and storage are essential in current information systems. Despite advancements in technology, the storage and transmission of massive volumes of data are serious tasks that must be considered. To overcome these issues, several compression techniques have been proposed to reduce the size of data by demanding a reduced area. Data Compression is the process of coding data with fewer bits than is usual. These techniques depend purely on the nature of data to be compressed. Several compression methods such as run length encoding, Huffman coding, Arithmetic Coding, LZW, JPEG, JPEG2000 have been proposed and implemented [21].

The proposed method uses the Lempel–Ziv–Markov chain algorithm (LZMA), which uses a constantly changing dictionary of strings to adaptively compress a stream of data by replacing common substrings by up to upto4Gb. It is essential to increase storage capacity, speed up file transfer, and reduce the cost incurred for hardware storage and bandwidth. The LZMA performs lossless data compression. Compared to other compression algorithms, LZMA has a higher compression ratio. A compressed stream is a stream of bits encoded using an adaptive binary-range coder. Although LZMA is primarily designed for compression, it can be adapted for use in encryption scenarios to enhance the data security. The fundamental idea is to utilize LZMA to compress plaintext data before encrypting them, resulting in an improved encryption efficiency.

Improved Encryption Efficiency: By applying LZMA compression to plaintext data before encryption, the resulting compressed data stream often contains a high frequency of recurring patterns and a reduced entropy [29]. Encryption algorithms operate more effectively when dealing with structured and predictable data rather than random data. The ability of LZMA to identify and eliminate redundancy helps create a more structured input for encryption, enhancing the overall efficiency.

1. Improved Energy Efficiency

By applying LZMA compression to plaintext data before encryption, the resulting compressed data stream often contains a high frequency of recurring patterns and a reduced entropy. Encryption algorithms operate more effectively when dealing with structured and predictable data rather than random data [30]. The ability of LZMA to identify and eliminate redundancy helps create a more structured input for encryption, enhancing the overall efficiency.

2. Enhanced Security

One of the critical advantages of incorporating LZMA [22] in encryption is increased security through obfuscation. The compressed output of LZMA does not reveal any identifiable patterns in the original plaintext. This feature makes it more difficult for potential attackers to gain insight into the nature of the data, thereby adding an additional security layer.

3. Bandwidth Optimization

In scenarios in which limited bandwidth is a concern, LZMA compression can significantly reduce the size of the data to be transmitted. A smaller data size directly translates to reduced bandwidth requirements and faster transmission times. By compressing the data before encryption, the LZMA aids in optimizing the utilization of network resources.

Trade-offs: It is important to consider trade-offs when integrating LZMA with encryption. The LZMA compression is a computationally intensive process that requires additional resources and time. Therefore, the computational overhead introduced by compression must be balanced by the potential benefits gained, in terms of enhanced encryption efficiency and security [31].

3. TRADE-OFFS

It is important to consider trade-offs when integrating LZMA with encryption. The LZMA compression is a computationally intensive process that requires additional resources and time. Therefore, the computational overhead introduced by compression must be balanced by the potential benefits gained, in terms of enhanced encryption efficiency and security.

3 Proposed modified play fair algorithm

The proposed algorithm generates 7 × 7 Playfair matrix to encrypt and decrypt ".txt" file. This matrix covers the upper- and lower-case letters, numbers, and special characters. A flow diagram of proposed model for encryption is divided into encryption input module and output module as shown in the Figs.1 and 2. P(x) is the original input and C(x) is the decrypted value. Similarly, the decryption processes also divided into decryption input and output module as shown in Figs. 3 and 4.

The proposed Playfair matrix encryption and decryption algorithm follows the following steps:

1) By dividing the plaintext into sections, each section can have two similar or dissimilar characteristics.
2) In case of similar characteristics, a filler character was added between them. The choice of filler used in the proposed algorithm was a random variable.
3) Encryption was performed following the rules of the traditional Playfair algorithm, except that the Playfair matrix table was generated based on the modified Septenary Quadrate Pattern, as shown in Table 2.
4) The number of filler characters and location of filler characters used in the cipher text were embedded in the cipher text. This enables decryption of the cipher text to obtain information regarding the fillers.
5) The size of the encrypted file increased because of the number of fillers added. Hence, to reduce the size of the encrypted file for transmission, the Lempel–Ziv–Markov chain algorithm (LZMA) compression algorithm, which retains the original size of the text file, is used for encryption.
6) The encoding schemes used are ANSI, UTF-8, UTE-16BE, UTF-16LE, UTF-32BE, UTF-32LE, and ASCII. To enhance the security, a random encoding scheme was chosen for each run.

Here is a high-level mathematical representation of the proposed process

Let the variables be:

- P: Plaintext
- C: Ciphertext
- F: Filler character
- M: Playfair matrix table
- E: Encryption function
- D: Decryption function
- LZMA: LZMA compression algorithm
- Enc: Encoding scheme

1) Divide the plaintext into sections

This step can be represented by dividing plain text P into pairs (P1, P2) of similar or dissimilar characters.

2) Add filler characters

If P1 and P2 have similar characteristics, the filler character F is inserted between them.

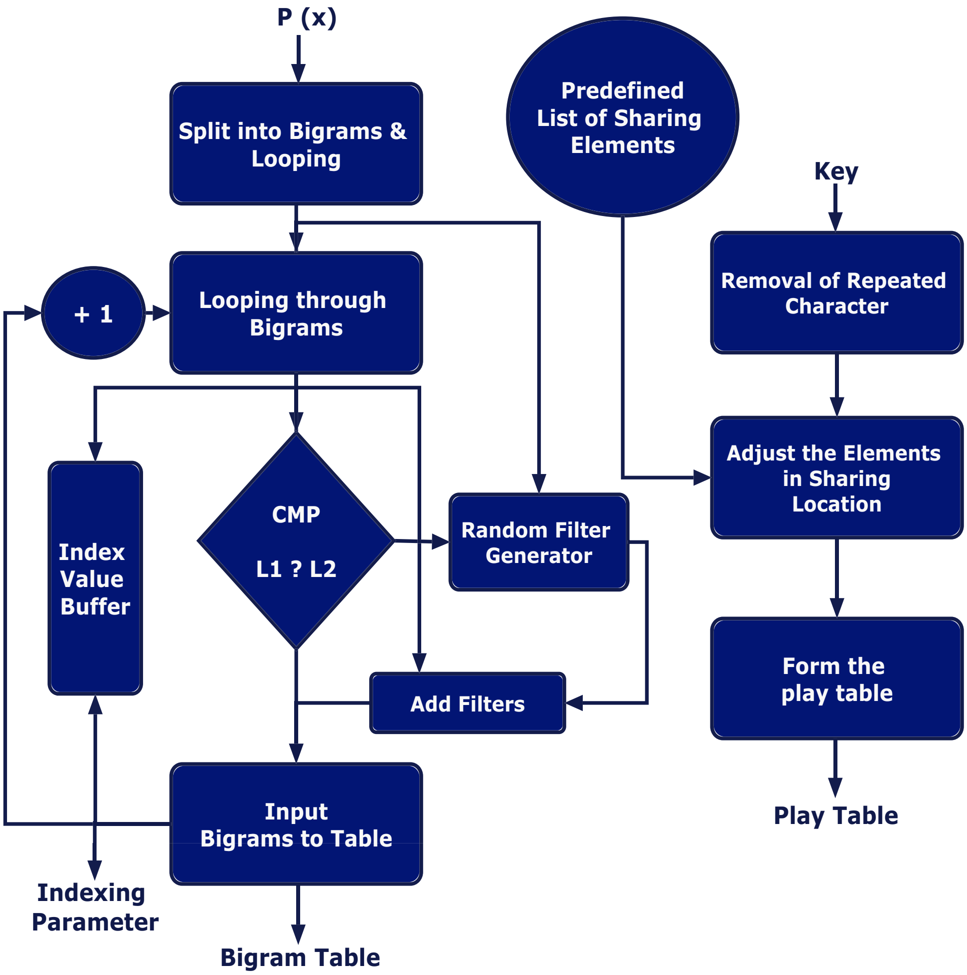


Fig. 1 Proposed encryption input module

3) Add filler characters
The encryption function E encrypts each pair of characters (P1, P2) using the modified Playfair algorithm with Playfair matrix table M as mentioned in Eq.1

$$C = E(P, M) \tag{1}$$

4) Embedding filler character information
The number of filler characters and their locations used in ciphertext.
5) Compression using lzma
The LZMA compression algorithm was applied to ciphertext C to reduce its size, while retaining the original text size as mentioned in Eq.2.

$$C\_compressed = LZMA, compress(C) \tag{2}$$

6) Coding scheme selection
A random encoding scheme, Enc, was chosen for each run to enhance security. The selected encoding scheme was used to encode the compressed ciphertext C _compressed as given in Eq.3.

$$C\_encoded = Enc.encode(C\_compressed) \tag{3}$$

The LZMA (Lempel-Ziv-Markov chain Algorithm) compression algorithm used in our algorithm involves several steps and equations. These equations and concepts provide a general understanding of LZMA compression algorithms. However, the LZMA is a complex algorithm with multiple variations and optimizations.
1) Probability computation
The LZMA algorithm utilizes a context modeling approach in which the probabilities of symbols are calculated based on previous symbols in the input data. The probability of a symbol 'S' occurring after context 'C' is calculated as given in the Eq.4.

$$P\_old(S|C) = \frac{Count(C, S)}{Count(C)} \tag{4}$$

2) Probability updating
As new symbols are processed, the probabilities are updated using adaptive models. One common method for

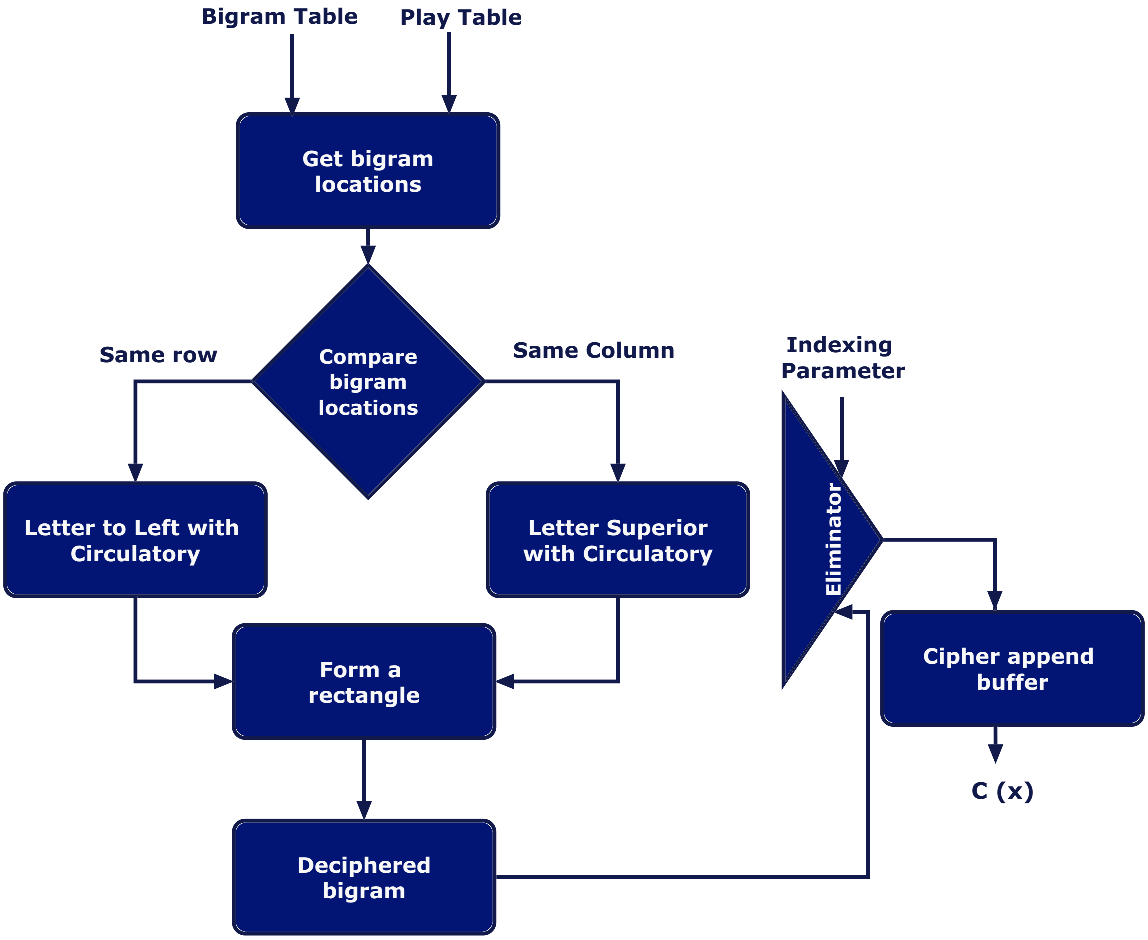


Fig. 2 Proposed Encryption output module

updating probabilities is called the “mixing probabilities” or "adaptive probability update" mentioned in Eq.5.

$$P\_new(S|C) = (1 - \alpha) * P\_old(S|C) + \alpha * P\_uniform \quad (5)$$

Here, P_old(S|C) is the previous probability of symbol ‘S' occurring after context 'C, ’ P_new(S|C) is the updated probability, P_uniform is a uniform probability distribution, and α is a mixing parameter between 0 and 1.

3) Encoding and decoding equations

The LZMA algorithm uses variable length encoding to represent symbols. The exact equations for encoding and decoding depend on the specific implementation of the LZMA algorithm, and may involve concepts such as range coding, bit manipulation, and data structures such as binary trees or hash tables.

4) Dictionary and matching

The LZMA algorithm employs a dictionary to store previously observed sequences of symbols. The matching process involves determining the longest match of symbols in the dictionary for the current input. The LZMA algorithm uses various data structures and algorithms to search and update the dictionary efficiently.

5) Compression ratio

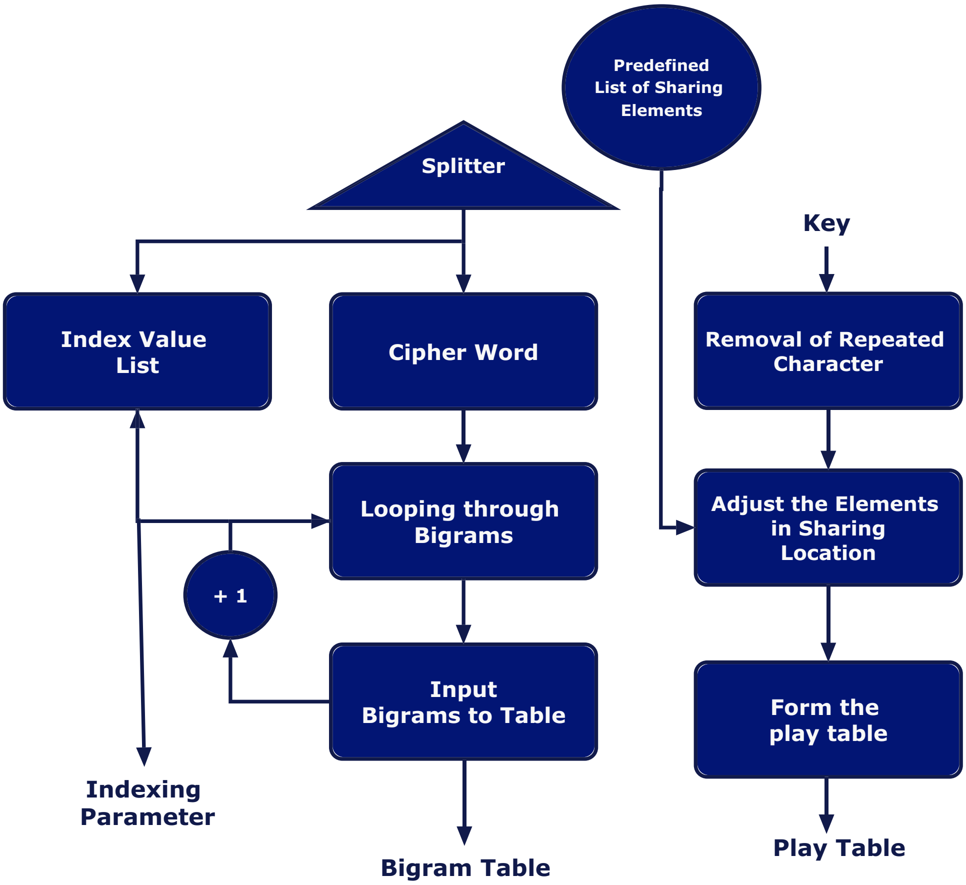
The compression ratio (CR) is a measure of the compression achieved by the LZMA algorithm. It is calculated as per Eq.6. Here, Compressed Size refers to the size of the compressed data, and Uncompressed Size refers to the size of the original uncompressed data.

$$CR = \frac{Compressed\,Size}{Uncompressed\,Size} \quad (6)$$

The most significant advantage of the proposed algorithm is that it can handle files of any size and size. The encryption and decryption times of a file depend on its size. Frequency analysis is typically used to break traditional play-fair algorithms. The proposed method was proven to withstand frequency analysis attacks because of its 7 × 7 matrix size, covering 98 characters with unpredictable random fillers.

The proposed algorithm is extremely sensitive to the keys. The randomness and predictability of filler

Fig. 3 Proposed Decryption Input Module



generation results in good key sensitivity. This is because the fillers are generated randomly, and information regarding the position of the fillers is highly random in cipher text. In addition, the selection of fillers is random for the entire text file and is not the same as that in the Playfair algorithm. This ensures that the proposed algorithm is secure against brute-force attacks. The lookup table computation for the proposed algorithm appears to be more complex, significantly increasing its efficiency.

The main restriction of the play fair cipher is that it encrypts only 26 characters. Special characters and numbers could not be encrypted using a 5 × 5 key matrix. Furthermore, keywords can take only 26 letters without duplicates. We propose a novel method for communicating data securely by using a 7 × 7 key matrix. The advantage of the proposed seasonal quadrate model is that it is designed to hold all possible characters present on a normal keyboard that is used for commercial purposes. This created a large key size and combination. In the proposed model, in addition to previous studies and matrices, a new pattern is created using a biograph to fill the elements in each location. The nomenclature for the generation of a pattern for a defined key is similar to that of the traditional 5 × 5 matrix; however, other elements in the key matrix are positioned randomly, which creates complexity in defining the elements of the key matrix. Hence, the key size and the combination of key matrices are significantly improved. The default key matrices are listed in table 1.

Once a key table is created, plain text encryption can be performed. During this process, the size of the cipher text was large because fillers were included. The addition of filler indices to encrypted message content serves as a source of confusion for intruders, thereby improving confidentiality. To reduce the size of the enciphered text file and induce security, the cipher text was compressed. Prior to compression encoding, the encoding type was randomized. This increases the redundancy in the cipher text, making it difficult for the intruder to access the data.

The encoding schemes used were ANSI, UTF-8, UTE-16BE, UTF-16LE, UTF-32BE, UTF-32LE, and ASCII. The compression algorithm reduces the size of the encrypted document by providing lossless compression with a high compression ratio. Likewise, the fillers used in this study were highly randomized, with a reduced pattern size compared with previous studies. The digraph character and substitution character accession times were significantly reduced.

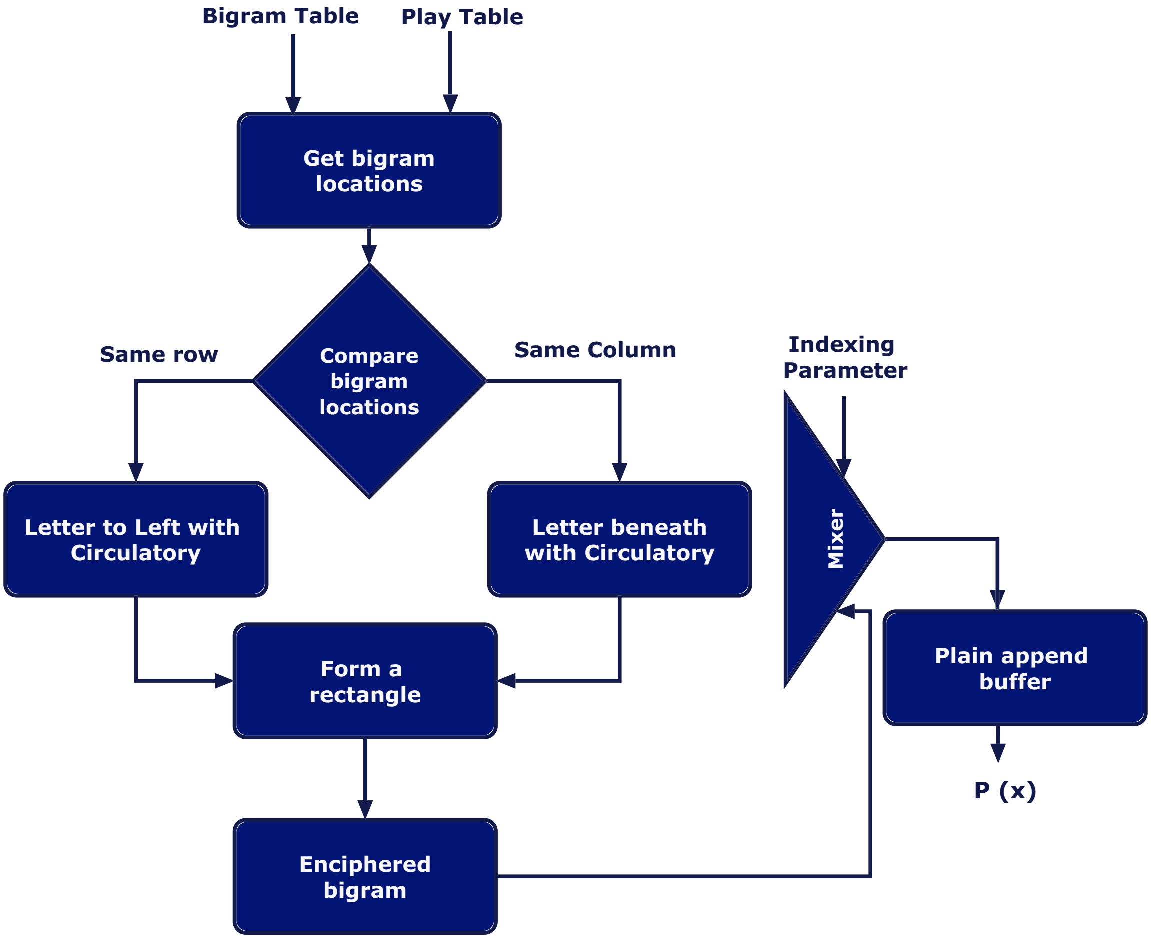


Fig. 4 Proposed Decryption Output Module

Table 1 Default septenary quadrate pattern

aA	bB	cC	dD	eE	fF	gG
hH	iI	jJ	kK	lL	mM	nN
oO	pP	qQ	rR	sS	tT	uU
vV	wW	xX	yY	zZ	\t	\n
01	23	45	67	89	!"	#$
%&	'(	)*	+,	-	/:	;<
=>	?@	[\	]^	_'	{|	}~

The proposed algorithm can also be applied to words, sentences, and paragraphs. The encrypted file was compressed using a random encoding scheme to ensure size reduction for easy transmission. This methodology uses the brute-force and avalanche effects. The avalanche effect is a desirable property of any cryptographic algorithm, that is, a small change in the plaintext (or key) should create a significant change in the ciphertext as mentioned in Eq.7.

$$\textit{Avalanche Effect} = \frac{\textit{Number of flipped bits in cipher text}}{\textit{Total number of bits in cipher text}} \tag{7}$$

The deciphered text is authentic since the fillers used are removed at the decryption.

The recorded outputs of the samples are presented in Table 2. The compression ratio, as given in Eq. 8, was approximately 0.4 to 0.8, depending on the size of the ciphertext. Using the proposed algorithm, a good compression ratio is achieved for cipher text with a size greater

Table 2 Compression ratio of encrypted message

S. no	Size of plain text (in KB)	Size of Key 1 (in character)	Size of Key 2 (in character)	Size of Cipher text (in KB)	Size of encoded Cipher text (in KB)	Compression ratio
1	3	25	80	5	3	0.4
2	10	25	80	13	7	0.46
3	20	25	80	23	10	0.565
4	50	25	80	57	22	0.614
5	100	25	80	108	27	0.75

than 10 KB. It was observed that when the size of the text file increased, the compression ratio improved proportionally. The compression ratio achieved here is similar to that of Ref.[24].

$$Compression\ ratio = \frac{Size\ of\ cipher\ text - size\ of\ encoded\ cipher\ text}{Size\ of\ Cipher\ text} \quad (8)$$

4 Results and discussion

All simulations of the modified Playfair algorithm were carried out on an Intel core i4, 64-bit processor with 4 GB RAM and a processor speed of 1.10 GHz. The algorithm was coded using Python3.6, using Spyder IDE. The efficiency and security of the proposed method are demonstrated using the compression ratio and avalanche percentage, as tabulated in Tables 3 and 4. The CIA triad emphasizes the policies to be followed for information security within an organization; hence, for a good avalanche effect, cipher text should always satisfy a high avalanche value [23].

The proposed algorithm encrypts plain text, and the encrypted plain text is compressed using the Lempel–Ziv–Markov chain compression algorithm (LZMA) for easy transfer of cipher text. The algorithm was evaluated for randomly generated 110 text files of varying sizes from 2 to 200 KB. All these files were encrypted with keys 1 and 2 with varying character lengths, and it was observed that the generated cipher text was not identical but was of the same size.

The proposed algorithm produced an avalanche effect of more than 65% [25]. The experiment produced a drastic change in the output, even for a single-bit change in the input. Even for the same key, a large mismatch in the cipher text was found when evaluated for 100 samples, as shown in Table 3.This can withstand any frequency attack.

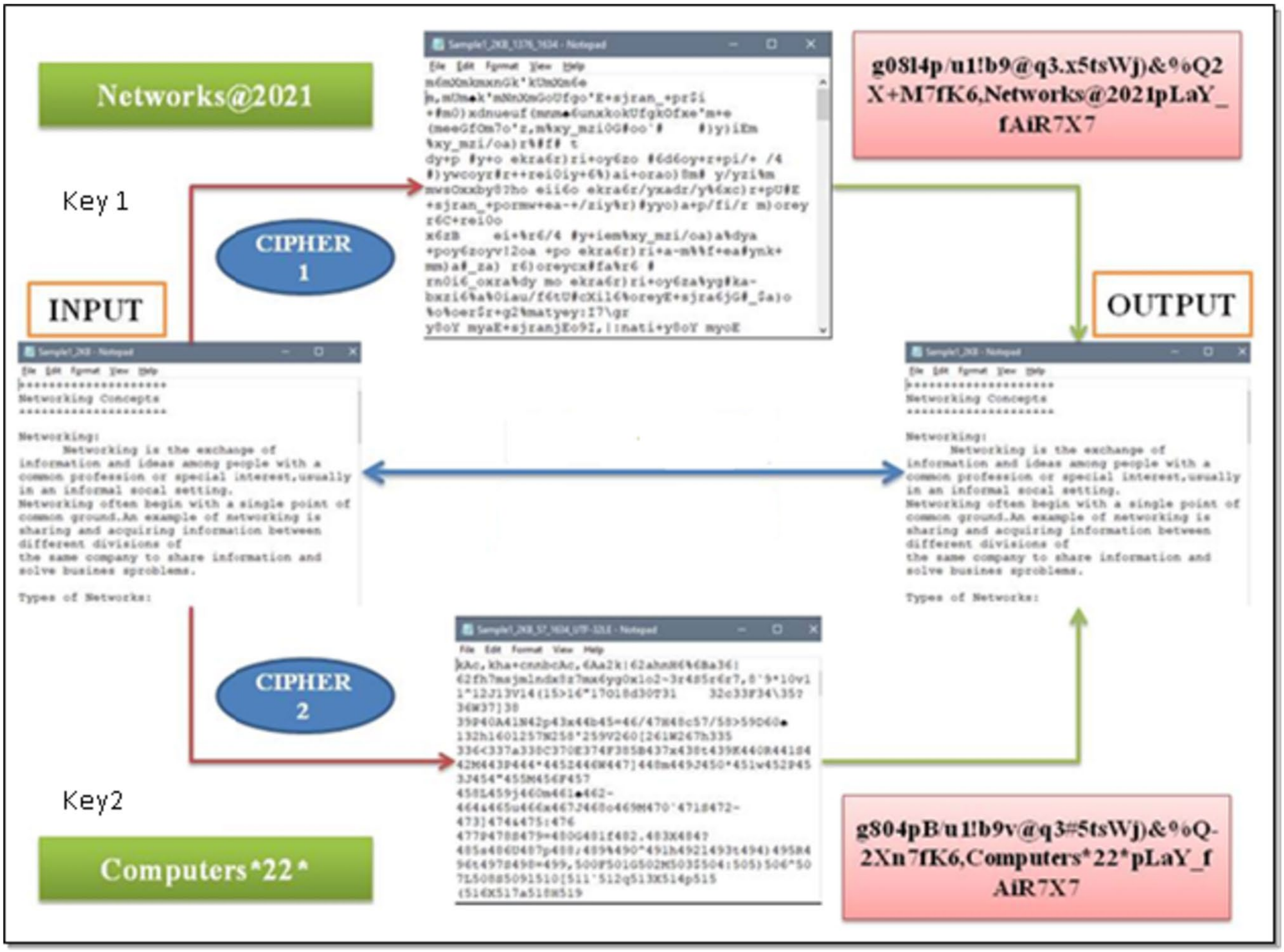
The sample output is shown in Fig. 5. This is a comparative output of the same plain text using different keys. Table 4 shows a comparison of the proposed algorithm with existing playfair techniques. Compared to existing techniques, our proposed method exhibits a 10% higher avalanche effect. Furthermore, a larger file size is considered. Larger file sizes owing to complex algorithms are inevitable.

Table 3 Results depicting the avalanche effect

Key 1	Key 2	Key Length	Plaintext length	Ciphertext Length	Mismatch length	Avalanche Percentage
Apples@123	Apples.123	10	14,179	33,836	25,752	76.10
21com017@sify.com	21com 015@sify.com	16	14,179	33,836	22,279	65.84
monkeysare1999@gmail.com	monkeysare 1999.gmail@ com	24	14,179	33,836	23,888	70.59
Network Security	Network/Security	16	14,179	33,836	28,063	82.93
Venrigbose0904	Venrigbose0004	14	14,179	33,836	26,747	79.04
grapes	GRAPES	6	14,179	33,836	30,382	89.79
GRapes	grAPES	6	14,179	33,836	29,614	87.52
GRApes	graPES	6	14,179	33,836	29,629	87.56
GRAPes	grapES	6	14,179	33,836	31,735	93.79
GRAPEs	grapeS	6	14,179	33,836	31,657	93.56
GRAPES	grapes	6	14,179	33,836	31,649	93.53
Keys are Same	Keys are Same	13	14,179	33,836	24,071	71.14
Keys are Same	Keys are Same	13	14,179	33,836	29,254	86.45

Table 4 Comparative Analysis

Ciphers	File size	Avalanche Effect	Repetition of character	Resilience towards frequency attack	Space requirement
4×4 Playfair cipher[3]	4 KB and 12 KB	46%–55%	15%	High	Less
8×8 Playfair cipher[26]	A sentence is considered	49.51%	Difference in bits 29 to 201	—	Not implemented in software
10×10 Playfair cipher[9]	A sentence is considered	0%–54.55%	Same case letter change produced 0% difference	Less	Not implemented in software but if implemented 10×10 will require high space
16×16 Playfair cipher[27]	10 to 40 characters	50%–60%	--	Brute force is high	Run time is 1 ms to 5 ms but lesser number of characters are considered
3D Playfair Cipher Algorithm[28]	A sentence is considered	54.17%	Number of bits flipped 6 to 260	High	–
Proposed method	3 KB to 100 KB	65%–94%	Mismatch length is above 20,000 bits	High	High


Networks@2021
Key 1
CIPHER 1
INPUT
g08l4p/u1!b9@q3.x5tsWj)&%Q2 X+M7fK6,Networks@2021pLaY_ fAiR7X7
OUTPUT
CIPHER 2
Key2
Computers*22*
g804pB/u1!b9v@q3#5tsWj)&%Q- 2Xn7fK6,Computers*22*pLaY_f AiR7X7

```
********************
Networking Concepts
********************

Networking:
        Networking is the exchange of
information and ideas among people with a
common profession or special interest,usually
in an informal socal setting.
Networking often begin with a single point of
common ground.An example of networking is
sharing and acquiring information between
different divisions of
the same company to share information and
solve busines sproblems.

Types of Networks:
```

```
********************
Networking Concepts
********************

Networking:
        Networking is the exchange of
information and ideas among people with a
common profession or special interest,usually
in an informal socal setting.
Networking often begin with a single point of
common ground.An example of networking is
sharing and acquiring information between
different divisions of
the same company to share information and
solve busines sproblems.

Types of Networks:
```

Fig. 5 Sample output for two different keys for the same plain text

Future research should focus on reducing the memory area in order to make it more suitable for IOT devices.

Tradational playfair algorithm works on a matrix size of 5×5 and with limited characters of 25, while this proposed model can handle characters in a 7×7 matrix of 98 characters. This enables text to be encrypted in variety of ways. Listed are the advantages of the proposed algorithm:

1. All the keyboard characters are included within the pattern.
2. The key size and key combinations are larger.
3. The fillers used are highly randomized when compared to traditional playfair algorithm.
4. The digraph character accession time and the substitution character accession time are highly reduced.
5. The Encryption and Decryption of a word, sentence and paragraphs are all possible.
6. The PlayFair cipher algorithm is enhanced using the proposed model with which text file encryption has been possible.
7. The addition of filler indices into the enciphered message content also serves as a source of confusion for the intruder.
8. The encrypted file is again compressed by using random encoding scheme to ensure size reduction for easy transmission.
9. The deciphered text is authentic since the fillers used are removed at the decryption.

5 Conclusion

In the modified Playfair algorithm, the traditional rules of the Playfair cipher are maintained, but the dimensions of the matrix are increased to 7 × 7. Unlike the traditional PlayFair cipher, there are no restrictions on the size of the keys used in this algorithm. This modification allows the algorithm to effectively handle text files of any size. During testing, the proposed algorithm demonstrated an avalanche effect ranging from 65 to 93.5%. The avalanche effect refers to the property of a cryptographic algorithm in which a small change in the input or key produces a significantly different output. A high avalanche effect indicates that the algorithm is robust and sensitive to changes in plaintext or key.

In addition, the algorithm achieved a compression ratio of 0.75 for a file size of 100 KB. This compression ratio signifies that the size of the cipher text is reduced, making it suitable for efficient file transfers. The LZMA compression algorithm is used in this process to further minimize the size of the cipher text. A notable advantage of the modified algorithm is its resistance to brute force attacks. Brute-force attacks involve trying all possible combinations of keys until the correct combination is obtained. Resistance of the algorithm to such attacks for the 49! Key combinations ensure the security of encrypted data. Moreover, the algorithm satisfies both the strict plaintext avalanche criterion and strict key avalanche criterion. These criteria evaluate the algorithm's ability to exhibit significant changes in the output cipher text when either the plaintext or the key is modified. By meeting these criteria, the algorithm ensures a high level of security and can be implemented as a security protocol in various applications including low-power embedded devices.

Acknowledgements We would like to thank all our universities, institutes, and organizations for facilitating their time and support in this study.

Author contributions NS, SK and HD conceived and designed the research, and NS, SK, RJ, BT, CV and HD wrote the first draft of the manuscript. KS, LD and LD all contributed in a substantial way to the writing process. All the authors revised the manuscript. All the authors read and approved the final manuscript.

Funding N/A.

Data availability Specific data set or data base are not used in the proposed algorithm.

Declarations

Conflict of interest The authors declare that they have no competing interests.

References

1. Abbott S, van Tilborg HCA (2001) Fundamentals of cryptology: a professional reference and interative Tutoria. Math Gaz 85(504):562. https://doi.org/10.2307/3621815
2. Albahrani EA, Maryoosh AA, Lafta SH, Block image encryption based on modified playfair and chaotic system, J Inf Secur Appl 51, 2020 https://doi.org/10.1016/j.jisa.2019.102445.
3. Mukherjee S, Chattopadhyay M, Lahiri A, Chattopadhyay S An efficient binary playfair algorithm using a 4×4 Playfair Key Matrix, Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 7564 LNCS, pp. 314–325, 2012, https://doi.org/10.1007/978-3-642-33260-9_27.
4. Singh K, Awasthi AK Quality, reliability, security and robustness in heterogeneous networks, 115, no. January. 2013.

5. Al-Kazaz NR, Teahan WJ (2018) An automatic cryptanalysis of playfair ciphers using compression. Int J Adv Comput Sci Appl 9(11):115–124. https://doi.org/10.14569/ijacsa.2018.0911105
6. Maha MM, Masuduzzaman M, Bhowmik A An effective modification of play fair cipher with performance analysis using 6 × 6 matrix, ACM Int. Conf. Proceeding Ser., no. January, 2020, https://doi.org/10.1145/3377049.3377085.
7. Deepa B, Maheswari V, Balaji V (1964) An efficient cryptosystem using playfair cipher together with graph labeling techniques. J Phys Conf Ser 2:2021. https://doi.org/10.1088/1742-6596/1964/2/022016
8. Manliclic GMM, Lamac KAR, Regala RC, MCR, Dioses RM (2023) Improving the extended 10 × 10 Polybius square key matrix for playfair, bifid, and polybius cipher, United International J Res Technol 04(07): 290–295
9. J. C. C. Ferrer, F. E. De Guzman, K. L. E. Gardon, R. J. R. Rosales, D. A. Dell Michael Badua, and D. R. Marcelo, "Extended 10 x 10 playfair cipher," 2018 IEEE 10th Int. Conf. Humanoid, Nanotechnology, Inf. Technol. Commun. Control. Environ. Manag. HNICEM 2018, pp. 1–4, 2019, https://doi.org/10.1109/HNICEM.2018.8666250.
10. Dhenakaran SS, Ilayaraja M (2012) Extension of playfair cipher using 16x16 matrix. Int J Comput Appl 48(7):37–41. https://doi.org/10.5120/7363-0192
11. N. Sharma, H. Meghwal, M. Mehta, and T. Kumar, "A Review on Playfair Substitution Cipher and Frequency Analysis Attack on Playfair," Proc. 2nd Int. Conf. Trends Electron. Informatics, ICOEI 2018, pp. 1–9, 2018, https://doi.org/10.1109/ICOEI.2018.8553837.
12. VM, Ayesha Sk (2018) A novel digital watermarking scheme for data authentication and copyright protection in 5G networks, Comput Electr Eng. 72: 589–605, https://doi.org/10.1016/j.compeleceng.2018.02.045/.
13. Andrew Onesimu J, Karthikeyan J, Eunice J, et al. (2022) Privacy preserving attribute-focused anonymization scheme for healthcare data publishing. IEEE Access. 10: 86979–86997. https://doi.org/10.1109/ACCESS.2022.3199433
14. Mahlake N, Mathonsi TE, Du Plessis D, Muchenje T (2023) A lightweight encryption algorithm to enhance wireless sensor network security on the internet of things. J Commun 18(1):47–57. https://doi.org/10.12720/jcm.18.1.47-57
15. Greeshmanth CR, Shah MA (2023) Novel secure data protection scheme using Martino homomorphic encryption, J Cloud Comput, 12(1) https://doi.org/10.1186/s13677-023-00425-7.
16. Srinivasarao AA, Tumma AY, Markapudi B, Chaduvula KA (2023) Smart Strategy for Data Hiding using Cryptography and Steganography, J Sci Ind Res 82(05), https://doi.org/10.56042/jsir.v82i05.1090.
17. Karthikayani K, Elumalai G, Jeyapiriya K, Southry SS, Gayathri S, Balamurugan A (2022) Performance analysis of various segmentation algorithms for microarray images. AIP Conf Proc 2518(1):070001. https://doi.org/10.1063/5.0103474
18. N. G. Goyal, S., Pacholi, B.S., Rao, B.A., Rai, S., Kini, "Parallel Message Encryption Through Playfair Cipher Using CUDA," in Evolution in Computational Intelligence. Advances in Intelligent Systems and Computing, 2021, pp. 519–526, doi: https://doi.org/10.1007/978-981-15-5788-0_50.
19. Amalia, M. A. Budiman, and R. Sitepu, "File text security using Hybrid Cryptosystem with Playfair Cipher Algorithm and Knapsack Naccache-Stern Algorithm," J. Phys. Conf. Ser., vol. 978, no. 1, 2018, doi: https://doi.org/10.1088/1742-6596/978/1/012114.
20. John Paul J, Jone AA, Martin Sagayam,"IoT based remote transit vehicle monitoring with seat availability display system,"Przeglad Elektrotechniczny. et al (2021) ISSN 0033–2097, R 97(5):140–145. https://doi.org/10.15199/48.2021.05.25
21. B. Carpentieri, "Efficient compression and encryption for digital data transmission," Secur. Commun. Networks, vol. 2018, 2018, doi: https://doi.org/10.1155/2018/9591768.
22. Shanthi, T., Ramprasath, M., Kavitha, A., & Muruganantham, T. (2023). Deep learning based autonomous transport system for secure vehicle and cargo atching. Intelligent Automation and Soft Computing, 35(1), 957–969. https://doi.org/10.32604/iasc.2023.027775
23. Dawson E, Gustafson H, Pettitt AN (1992) Strict Key Avalanche Criterion. Australas J Comb 6:147–153
24. J. L. Epiphany, "Hardware Implementation of LZMA Data Compression Algorithm," Int. J. Appl. Inf. Syst., no. March 2013, 2013.
25. Thabit F, Alhomdy S, Jagtap S (2021) Security analysis and performance evaluation of a new lightweight cryptographic algorithm for cloud computing. Glob Transitions Proc 2(1):100–110. https://doi.org/10.1016/j.gltp.2021.01.014
26. Gaurav Sharma, Priyanka Goyal, and Shivpratap Singh Kushwah, "Implementation of Modified Playfair CBC Algorithm," Int. J. Eng. Res., vol. V5, no. 06, pp. 679–684, 2016, doi: https://doi.org/10.17577/ijertv5is060631.
27. Marzan RM, Sison AM, Medina RP (2019) Randomness analysis on enhanced key security of Playfair cipher algorithm. Int J Adv Trends Comput Sci Eng 8(4):1248–1253. https://doi.org/10.30534/ijatcse/2019/34842019
28. Villafuerte RS, Sison AM, Medina RP (2019) An improved 3d playfair cipher key matrix with dual cipher block chaining method. Int J Sci Technol Res 8(10):1013–1018
29. J. Rajalakshmi K. Sumangali J. Jayanthi K.Muthulakshmi, Artificial Intelligence with Earthworm Optimization Assisted Waste Management System for Smart Cities. (2023). Global Nest Journal. https://doi.org/10.30955/gnj.004712, vol. 25, no. 4,pp. 190–197.
30. Gopi, R., Sheeba, R., Anguraj, K., Chelladurai, T., Alshahrani, H. M., Nemri, N., & Lamoudan, T. (2023). Intelligent Intrusion Detection System for industrial Internet of things environment. Computer Systems Science and Engineering, 44(2), 1567–1582. https://doi.org/10.32604/csse.2023.025216
31. Jamuna Rani, M. , & Vasanthanayaki, C. (2023). ELM-Based Shape Adaptive DCT Compression technique for underwater image compression. Computer Systems Science and Engineering, 45(2), 1953–1970. https://doi.org/10.32604/csse.2023.028713