Matrix Manipulations and Tridiagonal Structures New Schemes in Secure Data Encryption

Athar Kharal , Syed Ahtsham Ul Haq Bokhary * , Maha Mohammad Saeed

Posted Date: 31 October 2024

doi: 10.20944/preprints202410.2503.v1

Keywords: tridiagonal matrices; matrix manipulations; data encryption; symmetric key cryptography

Article

Matrix Manipulations and Tridiagonal Structures: New Schemes in Secure Data Encryption

Athar Kharal [1,†], Syed Ahtsham Ul Haq Bokhary [1,*,†] and Maha Mohammed Saeed [2,†]

[1] Centre for Advanced Studies in Pure and Applied Mathematics, Bahauddin Zakariya University, Multan - 60000, Pakistan
[2] Department of Mathematics, King Abdulaziz University, Jeddah, Saudi Arabia
* Correspondence: sihtsham@gmail.com; Tel.: +92 334 4141694
† These authors contributed equally to this work.

Abstract: This paper presents four novel schemes for secure data encryption utilizing mathematical transformations. The first scheme employs matrix manipulations and partitioning of numerical values derived from plain text characters, constructing a cipher through matrix additions. The second scheme introduces division operations and additional transformation layers for enhanced data obscurity. The third scheme incorporates tridiagonal matrices, creating a structured encryption process. The fourth scheme refines this approach with quotient-remainder operations and symmetric keys to construct a tridiagonal matrix cipher. Detailed algorithms for encryption and decryption are provided, with examples illustrating practical applications. This work contributes to mathematical cryptography by offering enhanced security methods for digital communication.

Keywords: tridiagonal matrices; matrix manipulations; data encryption; symmetric key cryptography

1. Introduction

Mathematical cryptography forms a crucial aspect of modern information security, employing advanced techniques to protect data against unauthorized access. The complexity and sophistication of cyber threats demand the development of robust cryptographic algorithms. This paper presents four innovative encryption schemes, each utilizing distinct mathematical transformations and methodologies to enhance data security.

The first scheme employs matrix manipulations and partitioning of numerical values derived from plain text characters, constructing a cipher through matrix additions. The second scheme builds on this foundation, introducing division operations and additional layers of transformation to further obscure the encrypted data. The third scheme incorporates tridiagonal matrices, creating a structured and secure encryption process. The fourth scheme refines this approach, using quotient-remainder operations and symmetric keys to construct a tridiagonal matrix cipher.

Each scheme is detailed, with step-by-step algorithms provided for both encryption and decryption processes. Examples illustrate the practical application of these methods, and a comprehensive discussion highlights the strengths and potential vulnerabilities of each approach. This paper aims to contribute to the field of mathematical cryptography by presenting these novel schemes, offering enhanced security for digital communication.

Paper Organization

This paper is organized as follows: Next Section 2 locates, links and orientates the present work in contemporary landscape of research in cryptography. Section 3 first presents the required preliminaries (3.1) and then Schemes 1 to 4 in subsections 3.2, 3.3, 3.4 and 3.5 . Section 3.6 presents the complexity analyses and finally Section 4 concludes the work.

2. Related Work

Cryptography's applications have garnered significant research attention. Lalitha and Vasu proposed a Python-based method for encoding and decoding text [1]. Amudha, Sagayaraj, and Sheela [2], along with Chowdhury, Ghosh, and Jana [3], explored techniques for securing communications.

 doi:10.20944/preprints202410.2503.v1

Kumari introduced a modified affine cipher algorithm for securing network communications, using matrices for data encryption and converting it into images [4]. Meenakshi et al. highlighted efficient domination and total magic labeling techniques for encryption and decryption [5]. Perera and Wijesiri discussed using matrices as secret keys in symmetric cryptography, enhancing data protection by converting plaintext into ciphertext [6].

The rise of edge computing has significantly transformed network management, facilitating the dispersal of services throughout the entire network rather than centralizing them within the cloud [7]. Tanguy Godquin et al. proposed a novel approach for deploying security services within IoT networks, considering the capabilities of individual devices [8]. By applying this approach to a smart city scenario, specifically employing IPsec services, the authors observed enhanced network security with minimal disruption to information flow and minimized deployment costs.

Privacy preservation in multi-agent systems (MASs) is paramount, particularly in the context of data exchange. Zewei Yang et al. tackled this issue by investigating privacy-preserving bipartite consensus control for discrete-time nonlinear MASs [9]. Employing the Paillier encryption scheme and event-triggered schemes, they ensured data confidentiality while maintaining communication efficiency. Theoretical analyses, supported by numerical examples, validated the effectiveness of their approach, offering insights into achieving bounded bipartite consensus in MASs.

The advent of quantum computing poses challenges to traditional cryptographic algorithms. Xiaohui Zhang et al. proposed leveraging topological graphic passwords for enhanced security against AI-equipped attacks and quantum threats [10]. Their study introduced new techniques utilizing graph colorings and labelings, offering insights into the design of secure encryption systems resilient to emerging technological threats.

Recognizing encrypted traffic poses challenges, particularly in the absence of prior label information. Ruipeng Yang et al. proposed a subspace clustering via graph auto-encoder network (SCGAE) to address this issue [11]. Their approach integrated graph encoder-decoder structures with self-supervised learning, yielding superior performance in identifying unknown encrypted traffic compared to benchmark models.

In conclusion, these studies collectively contribute to advancing the understanding and implementation of security measures, optimization techniques, and cryptographic systems in diverse network environments, addressing the evolving challenges posed by emerging technologies and dynamic network landscapes.

3. Main Work

In this section, the data is secured using the standard encoding (Table 1). From the set of values, matrices are constructed by using basic modular arithmetic. It is important to note that the elements of sets constructed in these algorithms can repeat and are ordered. We begin with the preliminaries first:

3.1. Preliminaries

Cryptography is the art of transforming original messages into an incomprehensible form, ensuring that only authorized parties can access the information. This transformation is achieved through mathematical techniques and algorithms designed to encrypt (encode) data securely. The entire process of secure communication is facilitated by a framework known as a cryptosystem, which comprises five essential components: the key, the encryption algorithm, the decryption algorithm, the cipher-text, and the plain-text.

The plain text is the original message sent by the sender, while the cipher text is the secret message or the encrypted version of the plain text. The encryption process involves converting the plain text into cipher text using a cryptographic algorithm and a key. Conversely, decryption is the process of converting the cipher text back into plain text using a key. The cryptographic algorithm is a mathematical algorithm employed to perform both encryption and decryption. A key, which is

 doi:10.20944/preprints202410.2503.v1

a secret value used by cryptographic algorithms, is crucial for encrypting and decrypting data. To ensure the security of the encrypted information, keys must be kept confidential.

Cryptographic techniques can be broadly classified into two categories based on key distribution: symmetric key cryptography and asymmetric key cryptography.

Symmetric key cryptography, also known as secret key cryptography, employs the same key for both encryption and decryption. This method involves using a single encryption key for electronic communication, making use of a secret key and numerical computation to secure the data. In symmetric key cryptography, both participants use the same key K to encrypt and decrypt data, thereby protecting it from unauthorized access.

On the other hand, asymmetric key cryptography, also known as public key cryptography, utilizes a pair of keys: a public key and a private key. In this method, the public key is used for encryption, while the private key is used for decryption. These keys are essentially large, non-distinguishable numbers that have been generated to work together. This form of cryptography allows for more secure and flexible communication, as the public key can be freely distributed while the private key remains confidential.

By combining the strengths of symmetric and asymmetric key cryptography, modern cryptographic systems can provide robust security for a wide range of applications, ensuring the confidentiality, integrity, and authenticity of the information being transmitted.

3.1.1. Encoding Tables

There are two types of encoding table used in this work to encrypt/decrypt the given data. One is the standard encoding table and the other is the Pythagorean encoding table.

Table 1. Standard Encoding Table

A	B	C	D	E	F	G	H	I	J	K	L	M
1	2	3	4	5	6	7	8	9	10	11	12	13
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
14	15	16	17	18	19	20	21	22	23	24	25	26

Table 2 displays the Pythagorean numbers corresponding to the alphabets and Table 3 illustrates how the numbers are connected to the alphabets in order to allow for uniqueness in vertex labeling of a graph.

Table 2. Pythagorean numbers corresponding to each alphabet.

	1	2	3	4	5	6	7	8	9
1	A	B	C	D	E	F	G	H	I
2	J	K	L	M	N	O	P	Q	R
3	S	T	U	V	W	X	Y	Z	

Table 3. Adjusted table used in encryption/decryption scheme

A	B	C	D	E	F	G	H	I	J	K	L	M
11	21	31	41	51	61	71	81	91	12	22	32	42
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
52	62	72	82	92	13	23	33	43	53	63	73	83

The details of the encryption and decryption algorithm are listed below.

3.2. Scheme 1: Novel Cryptographic Protocol Using Matrix Conversions and Partitioned Values for Enhanced Security

Following algorithm initiates with the plain text, partitioning and interchanging values to form distinct sets. These sets are then ingeniously transformed into matrices, which through specific

 doi:10.20944/preprints202410.2503.v1

mathematical manipulations, yield the final cipher text. The decryption process mirrors this complexity, systematically reversing the operations to retrieve the original message. Each step, grounded in mathematical precision, contributes to a resilient cryptographic protocol capable of withstanding potential security breaches.

Scheme 1 : Description

Encryption

1. Let P be the plain text of length n.
2. Draw the Encryption Table 3.
3. Let A_1 be the set of values corresponding to the alphabets.
4. Partition A_1 into two sets such that Q_1 is the set of values in A_1 that are at ten's place and R_1 is the set of values in A_1 that are at unit's place. It is important to note that all the numbers in A_1 have two digits.
5. Construct a new set A_2 from the set A_1 by interchanging the values.
6. Similarly, partition A_2 in two parts such that Q_2 is the set of values in A_2 that are at ten's place and R_2 is the set of values in A_2 that are at unit's place.
7. Construct a matrix M_1 of order n with each row R_2.
8. Construct a diagonal matrix M_2 of order n with diagonal entries from the set Q_2.
9. Adding matrices M_1 and M_2 to obtain the matrix M.
10. Construct a list according to each column of the matrix M. This list of n^2 element is our cipher.

Decryption

1. A list of numbers is received by the receiver.
2. Count the element of the cipher and by taking the square root of the obtained number. The length of the original plain text is decrypted.
3. Construct the matrix $decM$ from the cipher according to each column.
4. Arrange the diagonal entry in an array and denote it by $decD_1$.
5. Construct an array $decR_2$ by taking 1^{st} element from 21 entry of the matrix $decM$ and the remaining $n-1$ elements are the entries of first row of $decM$ starting from 2^{nd} position to the n^{th} position.
6. Subtract $decR_2$ from $decD_1$ to obtained the $decQ_2$.
7. Multiplying $decQ_2$ by 10 and adding to the corresponding element of $decR_2$. We obtained the set $decA_2$.
8. Divide each element of $decA_2$ by 10 to obtained the sets $decQ_1$ and $decR_1$ which are respectively the set of quotients and the set of remainders.
9. Multiply each element of $decR_1$ by 10 and adding to the corresponding element of $decQ_1$ to obtained the set $decA_1$.
10. Find the corresponding alphabets from the Table 3.
11. The original message is decrypted.

Example 1. ***Encryption***

Let the plain text be ***MATRIX*** *of length* 6*. Then we have and thus*

$$A_1 = [42, 11, 23, 92, 91, 63] \qquad Q_1 = [4, 1, 2, 9, 9, 6] \quad R_1 = [2, 1, 3, 2, 1, 3]$$
$$A_2 = [24, 11, 32, 29, 19, 36] \qquad Q_2 = [2, 1, 3, 2, 1, 3] \quad R_2 = [4, 1, 2, 9, 9, 6]$$

 doi:10.20944/preprints202410.2503.v1

Algorithm 1: Scheme 1

Encryption
INPUT : $plaintext \leftarrow$ string as received from user
OUTPUT $cypher \leftarrow$ list of integers
:
A_1 Table 3 equivalents of $plaintext$
$n \leftarrow length(plaintext)$
$Q_1, Q_2, R_1, R_2, A_2 \leftarrow [\ ]$
FOR i *from* 1 *to* n **DO**
  $Q_1 \leftarrow Q_1.append(quotient(A_1[i], 10))$
  $R_1 \leftarrow R_1.append(remainder(A_1[i], 10))$
  $A_2 \leftarrow A_2.append(10 \cdot R_1[i] + Q_1[i])$
  $Q_2 \leftarrow Q_2.append(quotient(A_2[i], 10))$
  $R_2 \leftarrow R_2.append(remainder(A_2[i], 10))$
END
$M_1 \leftarrow [\ ]_{n \times n}$
FOR i *from* 1 *to* n **DO**
  FOR j *from 1 to* n **DO**
    $M_1[i, j] \leftarrow 0 \cdot i + R_2[j]$
  END
END
$M_2 \leftarrow DiagonalMatrix(Q_2)$
$M \leftarrow M_1 + M_2$ $cypher \leftarrow convert(M, flatList)$
RETURN $cypher$

Decryption
INPUT : $cypher$ :: list of integers
OUTPUT $plaintext \leftarrow$ string as recovered
:
$decM \leftarrow reshape(cypher, n \times n)$
$decMM \leftarrow [\ ]$
FOR i *from 1 to* n **DO**
  FOR j *from 1 to* n **DO**
    IF $i \neq j$ **THEN**
      $decMM[i, j] \leftarrow decM[i, j]$
    ELSE IF $i \neq 1$ **THEN**
      $decM[i, j] \leftarrow decM[1, 1]$
    ELSE
      $decMM[i, j] \leftarrow decM[2, 1]$
  END
END
$decQQ \leftarrow diagonal(decM)$
$decR_2 \leftarrow convert(decM, list)$
$decQ_2 \leftarrow convert(decQQ, list) - decR_2$
FOR i *from 1 to* n **DO**
  $decA_2 \leftarrow decA_2.append(n \cdot decQ_2[i] + decR_2[i])$
  $decQ_1 \leftarrow decQ_1.append(quotient(decA_2, 10))$
  $decR_1 \leftarrow decR_1.append(remainder(decA_2, 10))$
  $decA_1 \leftarrow decA_1.append(10 \cdot decR_1[i] + decQ_1[i])$
END
$plaintext \leftarrow$ *Table 3* equivalents of $decA_1$
RETURN $plaintext$

 doi:10.20944/preprints202410.2503.v1

Construct a matrix M_1 of order n with each row R_2, and another diagonal matrix M_2 of order n with diagonal entries from the set Q_2. Adding M_1 and M_2 yields our matrix M as shown below:

$$\overbrace{\begin{pmatrix} 4 & 1 & 2 & 9 & 9 & 6 \\ 4 & 1 & 2 & 9 & 9 & 6 \\ 4 & 1 & 2 & 9 & 9 & 6 \\ 4 & 1 & 2 & 9 & 9 & 6 \\ 4 & 1 & 2 & 9 & 9 & 6 \\ 4 & 1 & 2 & 9 & 9 & 6 \end{pmatrix}}^{M_1} + \overbrace{\begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{pmatrix}}^{M_2} \rightarrow \overbrace{\begin{pmatrix} 6 & 1 & 2 & 9 & 9 & 6 \\ 4 & 2 & 2 & 9 & 9 & 6 \\ 4 & 1 & 5 & 9 & 9 & 6 \\ 4 & 1 & 2 & 11 & 9 & 6 \\ 4 & 1 & 2 & 9 & 10 & 6 \\ 4 & 1 & 2 & 9 & 9 & 9 \end{pmatrix}}^{M}$$

The ordered list of 36 elements

$$[6,4,4,4,4,4,1,2,1,1,1,1,2,2,5,2,2,2,9,9,9,11,9,9,9,9,9,9,10,9,6,6,6,6,6,9]$$

is our cipher.

Decryption *Chopping the received cipher list into chunks of size $\sqrt{\text{length of list}} = \sqrt{36} = 6$ each and stacking to get the matrix*

$$decM = \begin{pmatrix} 6 & 1 & 2 & 9 & 9 & 6 \\ 4 & 2 & 2 & 9 & 9 & 6 \\ 4 & 1 & 5 & 9 & 9 & 6 \\ 4 & 1 & 2 & 11 & 9 & 6 \\ 4 & 1 & 2 & 9 & 10 & 6 \\ 4 & 1 & 2 & 9 & 9 & 9 \end{pmatrix}$$

*Extract diagonal of decM and recover $decR_2$, respectively as $decD_1 = [6,2,5,11,10,9]$ $decR_2 = [4,1,2,9,9,6]$. Subtract $decR_2$ from $decD_1$ to get $decQ_2 = [2,1,3,2,1,3]$. Multiplying $decQ_2$ by 10 and adding to the corresponding element of $decR_2$ to gives $decA_2 = [24,11,32,29,19,36]$. Divide each element of $decA_2$ by 10 to get $decQ_1 = [2,1,3,2,1,3]$ and $decR_1 = [4,1,2,9,9,6]$. Multiplying each element of $decR_1$ by 10 and adding to the corresponding element of $decQ_1$ to obtain $decA_1 = [42,11,23,92,91,63]$ and using Table 3 original text is decrypted as **MATRIX**.*

3.3. Scheme 2: Two-Layer Cryptographic Scheme Using Symmetric Key Transformation and Encoding

In this scheme (Scheme 2), the encryption algorithm initiates similarly to Scheme 1 but introduces a more elaborate series of steps involving additional partitions and transformations. This scheme enhances security by employing multiple layers of encoding and manipulation, which are further reinforced by the use of symmetric keys. The decryption algorithm, though more complex, effectively reverses these intricate steps to retrieve the original plain text, ensuring both security and accuracy in the encryption process.

Scheme 2 : Description

Encryption

1. Do 5 steps of scheme 1.
2. Divide each element of A_2 by 26 to get the sets Q_2 and R_2 that are the sets of quotient and the remainders, respectively.
3. Find the alphabets from the Table 1 corresponding to each element of the R_2. Let the plain text be $cipher1$.
4. From Table 3, find the values corresponding to each alphabets of $cipher1$ and let A_3 be the ordered set containing all the values.

 doi:10.20944/preprints202410.2503.v1

5. Divide each element of A_3 by 10 and let Q_3 and A_3 are the sets of quotient and remainder, respectively.
6. Multiplying R_3 by 10 and then add to the corresponding component of Q_3. The resulting values will be stored in an ordered set A_4.
7. Add the corresponding element of A_3 and A_4 and denote the resulting set by A_5.
8. Divide each element of A_5 by n and let Q_4 and R_4 are the respective sets of quotients and remainders, respectively.
9. Back to scheme 1

Decryption

1. Let the receiver has received the message as a string of numbers together with two sets of symmetric keys Q_2 and R_3.
2. Count the element of the cipher and by taking the square root of the obtained number. The length of the original plain text is decrypted.
3. Construct the matrix $decM$ from the cipher according to each column.
4. Arrange the diagonal entry in an array form and denote by $decD_2$.
5. Construct an array $decR_4$ by taking 1^{st} element of the matrix $decM$ from the entry 21 and the remaining $n-1$ elements of the matrix $decM$ from row R_1 starting from 2^{nd} position to the n^{th} position.
6. Subtract $decR_4$ from $decD_2$ to obtained the $decQ_4$.
7. Multiplying $decQ_4$ by n and adding to the corresponding element of $decR_4$. We obtained the set $decA_5$.
8. Divide $decA_5$ by 11 to obtained the set decC that are the set of quotient. And then subtract it from the symmetric key R_3 to obtained the set $decQ_3$.
9. Multiplying each element of $decQ_3$ by 10 and add it to the corresponding element of the symmetric key R_3. Let us denote the resulting set as $decA_3$.
10. Find the alphabet corresponding to each value of $decA_3$ from the Table 3.
11. Find the values corresponding to each alphabet from the Table 1 to get $decR_2$.
12. Multiply each component of the symmetric key Q_2 by 26 and add in the previous set which is $decR_2$ and then subtract 1 from each element to decrypt A_2.
13. Divide each element of $decA_2$ by 10 to decrypt the sets Q_1 and R_1 which are the sets of quotients and remainders, respectively.
14. Multiply each element of $decR_1$ by 10 and then add them to the corresponding element of $decQ_1$ to decrypt A_1.
15. Find the alphabet corresponding to each element of $decA_1$ from 3.
16. The message is decrypted.

Example 2. ***Encryption*** *Let* $P =$ ***MATRIX*** *be the plain-text of length* 6. *Then we have* $A_1 = [42, 11, 23, 92, 91, 63]$, $Q_1 = [4, 1, 2, 9, 9, 6]$*and* $R_1 = [2, 1, 3, 2, 1, 3]$. *Constructing a new set* A_2 *from the set* A_1 *by interchanging the values and computing its quotients and remainders after dividing by 26 gives:* $A_2 = [24, 11, 32, 29, 19, 36]$, $Q_2 = [0, 0, 1, 1, 0, 1]$ *and* $R_2 = [24, 11, 6, 3, 19, 10]$. *Finding the alphabets from the Table 1 corresponding to each element of the set* R_2 *produces* $Cipher1 = YLGDTK$. *Using Table 3, for* $Cipher1$ *gives* $A_3 = [73, 32, 71, 41, 23, 22]$, $Q_3 = [7, 3, 7, 4, 2, 2]$ *and* $R_3 = [3, 2, 1, 1, 3, 2]$. *Multiplying* R_3 *by* 10 *and then adding to the corresponding component of* Q_3 *yields* $A_4 = [37, 23, 17, 14, 32, 22]$. *Add the set* A_3 *and* A_4 *we have*

 doi:10.20944/preprints202410.2503.v1

Algorithm 2: Scheme 2

Encryption
INPUT : $plaintext \leftarrow$ string as received from user
OUTPUT $cypher \leftarrow$ list of integers
:
A_1 Table 3 equivalents of $plaintext$
$n \leftarrow length(plaintext)$
$Q_1, Q_2, R_1, R_2, A_2 \leftarrow [\ ]$
FOR i *from* 1 *to* n **DO**
  $Q_1[i] \leftarrow quotient(A_1[i], 10)$
  $R_1[i] \leftarrow remainder(A_1[i], 10)$
  $A_2[i] \leftarrow 10 \cdot R_1[i] + Q_1[i]$
  $Q_2[i] \leftarrow quotient(A_2[i], 10)$
  $R_2[i] \leftarrow remainder(A_2[i], 10)$
END
$cypher_1 \leftarrow$ alphabets according to Table 1
$A_3 \leftarrow$ integer equivalents of $cypher1$ as per Table 3
FOR i *from* 1 *to* n **DO**
  $Q_3[i] \leftarrow quotient(A_3[i], 10)$: $R_3[i] \leftarrow remainder(A_3[i], 10)$
  $A_4[i] \leftarrow 10 \cdot R_3[i] + Q_3[i]$: $A_5[i] \leftarrow 10 \cdot A_3[i] + A_4[i]$
  $Q_4[i] \leftarrow quotient(A_5[i], n)$: $R_4[i] \leftarrow remainder(A_5[i], n)$
END
FOR i *from 1 to* n **DO**
  FOR j *from 1 to* n **DO**
    IF $i \neq j$ **THEN**
      $M[i, j] \leftarrow 0 \cdot i + R_4[j]$
    ELSE
      $M[i, j] \leftarrow Q_4[i]$
  END
END
$cypher \leftarrow convert(M, FlatList)$ **RETURN** $cypher$

Decryption
INPUT : $cypher$ :: list of integers, R_3, Q_2, n
OUTPUT $plaintext \leftarrow$ string as recovered
:
$decM \leftarrow reshape(cypher, n \times n)$
FOR i *from 1 to* n **DO**
  FOR j *from 1 to* n **DO**
    IF $i \neq j$ **THEN**
      $decMM[i, j] \leftarrow decM[i, j]$
    ELSE IF $i \neq 1$ **THEN**
      $decM[i, j] \leftarrow decM[1, 1]$
    ELSE
      $decMM[i, j] \leftarrow decM[2, 1]$
  END
END
$decQQ \leftarrow diagonal(decM)$: $decTQ \leftarrow transpose(convert(decQQ, Matrix, n \times 1))$
$decR_4 \leftarrow convert(decMM[1], list)$: $decQ_4 \leftarrow convert(decTQ, list) - decR_4$
$decA_5 \leftarrow n \cdot decQ_4 + decR_4$
$decC \leftarrow quotient(decA_5, 11)$: $decQ_3 \leftarrow decC - R_3$
$decA_3 \leftarrow 10 \cdot decQ_3 + R_3$: $decCypher1 \leftarrow$ Table 3$(decA_3)$
$decR_2 \leftarrow$ Table 1 : $decA_2 \leftarrow 26 \cdot Q_2 + decR_2$
$decQ_1 \leftarrow quotient(decA_2, 10)$: $decR_1 \leftarrow remainder(decA_2, 10)$
$decA_1 \leftarrow 10 \cdot decR_1 + decQ_1$
$plaintext \leftarrow Table3$ equivalents of $decA_1$
RETURN $plaintext$

 doi:10.20944/preprints202410.2503.v1

A_5 and consequently division of A_5 by 6 yields Q_4 and R_4 as follows: $A_5 = [110, 55, 88, 55, 55, 44]$, $Q_4 = [18, 9, 14, 9, 9, 7]$ *and* $R_4 = [2, 1, 4, 1, 1, 2]$. *Matrix M is obtained as follows:*

$$\overbrace{\begin{pmatrix} 2 & 1 & 4 & 1 & 1 & 2 \\ 2 & 1 & 4 & 1 & 1 & 2 \\ 2 & 1 & 4 & 1 & 1 & 2 \\ 2 & 1 & 4 & 1 & 1 & 2 \\ 2 & 1 & 4 & 1 & 1 & 2 \\ 2 & 1 & 4 & 1 & 1 & 2 \end{pmatrix}}^{M_1} + \overbrace{\begin{pmatrix} 18 & 0 & 0 & 0 & 0 & 0 \\ 0 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 14 & 0 & 0 & 0 \\ 0 & 0 & 0 & 9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7 \end{pmatrix}}^{M_2} \rightarrow \overbrace{\begin{pmatrix} 20 & 1 & 4 & 1 & 1 & 2 \\ 2 & 10 & 4 & 1 & 1 & 2 \\ 2 & 1 & 18 & 1 & 1 & 2 \\ 2 & 1 & 4 & 10 & 1 & 2 \\ 2 & 1 & 4 & 1 & 10 & 2 \\ 2 & 1 & 4 & 1 & 1 & 9 \end{pmatrix}}^{M}$$

List the element of the matrix as follows: $[20, 2, 2, 2, 2, 2, 1, 10, 1, 1, 1, 1, 4, 4, 18, 4, 4, 4, 1, 1, 1, 10, 1, 1, 1, 1, 1, 1, 10, 1, 2, 2, 2, 2, 2, 9]$. *This list of* 36 *elements is our cipher. For decryption, on receiving the 36 element list cipher, Q_2 and R_3, construct the matrix*

$$decM = \begin{pmatrix} 20 & 1 & 4 & 1 & 1 & 2 \\ 2 & 10 & 4 & 1 & 1 & 2 \\ 2 & 1 & 18 & 1 & 1 & 2 \\ 2 & 1 & 4 & 10 & 1 & 2 \\ 2 & 1 & 4 & 1 & 10 & 2 \\ 2 & 1 & 4 & 1 & 1 & 9 \end{pmatrix}$$

and extract its diagonal $decD_2 = [20, 10, 18, 10, 10, 9]$. *Next, construct an array* $decR_4 = [2, 1, 4, 1, 1, 2]$ *by taking* 1^{st} *element of the matrix decM from the entry* 21 *and the remaining* $n - 1$ *elements of the matrix decM from row* R_1 *starting from* 2^{nd} *position to the* n^{th} *position. Subtract* $decR_4$ *from* $decD_2$ *and multiplying* $decQ_4$ *by* 6 *and adding to the corresponding element of* $decR_4$ *one gets:* $decQ_4 = [18, 9, 14, 9, 9, 7]$ *and* $decA_5 = [110, 55, 88, 55, 55, 44]$

Divide $decA_5$ *by 10 to obtain the* $decC = [10, 5, 8, 5, 5, 4]$ *and then subtract it from the symmetric key* R_3 *to obtain the set* $decQ_3 = [7, 3, 7, 4, 2, 2]$. *Now get* $decA_3 = [73, 32, 71, 41, 23, 22]$ *by multiplying each element of* $decQ_3$ *by 10 and add it to the corresponding element of the symmetric key* R_3. *Using Table 3 over* $decA_3$ *gives* $YLGDTK$. *Further use of Table 1 yields* $decR_2 = [24, 11, 6, 3, 19, 10]$.

Multiplying each component of the symmetric key Q_2 *by 26 and adding in* $decR_2$ *and then subtract 1 from each element to obtain* $decA_2 = [24, 11, 32, 29, 19, 36]$. *Divide each element of* $decA_2$ *by* 10 *to decrypt the sets as* $Q_1 = [2, 1, 3, 2, 1, 3]$ *and* $R_1 = [4, 1, 2, 9, 9, 6]$. *Next, multiply each element of* $decR_1$ *by* 10 *and then add them to the corresponding element of* $decQ_1$ *to get the set* $decA_1 = [42, 11, 23, 92, 91, 63]$ *which by use of Table 3 yields the final decrypted message as* ***MATRIX***.

3.4. Scheme 3: Enhanced Cipher-Text Length Optimisation and Compact Keys for Two-Layer Cryptography Scheme

The algorithm of Scheme 3 integrates advanced mathematical operations to create a cipher from the plain text. The detailed process involves partitioning values, interchanging digits, and constructing tridiagonal matrices. These steps are designed to transform the plain text into an encrypted form that is challenging to decipher without the correct keys. Furthermore, the decryption algorithm ensures that the original message can be accurately reconstructed, maintaining the integrity of the data. The systematic approach of Scheme 3, combined with its use of tridiagonal matrices and symmetric keys, marks a significant improvement over previous schemes, offering enhanced security and complexity.

 doi:10.20944/preprints202410.2503.v1

Scheme 3 : Description

Encryption

1. Let P be the plain text of length n.
2. Draw the Encryption Table 3.
3. Let A_1 be the set of values corresponding to the alphabets.
4. Partition the values of A_1 in two parts such that Q_1 is the set of values in A_1 that are at ten's place and R_1 is the set of values in A_1 that are at unit's place.
5. Construct a new set A_2 from the set A_1 by interchanging the values.
6. Divide each element of A_2 by 26 to get the sets Q_2 and R_2 that are the sets of quotients and the remainders, respectively.
7. Find the alphabets from the Table 1 corresponding to each element of the set R_2. Let the plain text be $cipher1$.
8. From Table 3, find the values corresponding to each alphabets of $cipher1$ and let A_3 be the ordered set containing all the values.
9. Divide each element of A_3 by 10 and let Q_3 and A_3 are the sets of quotient and remainder, respectively.
10. Multiplying R_3 by 10 and then add to the corresponding component of Q_3. The resulting values will be stored in an ordered set A_4.
11. Add the corresponding element of A_3 and A_4 and denote the resulting set by A_5.
12. Divide each element of A_5 by n and let Q_4 and R_4 are the respective sets of quotients and remainders, respectively.
13. Construct the cipher as follows:
    - Construct the cipher as tridiagonal matrix with main diagonal set Q_4.
    - The lower sub-diagonal set as entries from the first $n-1$ elements of the set R_3.
    - The entries of the upper sub-diagonal are the last $n-1$ elements of the set Q_2.
14. The cipher text is obtained.

Decryption

1. Let the receiver has received the message as a matrix of order $n \times n$ together with set of symmetric keys Q_3 and n^{th} and 1^{st} entries of R_3 and Q_2 respectively.
2. Arrange the diagonal entry in a set to decrypt $decQ_4$.
3. Arrange the lower sub-diagonal entry of matrix received as first $n-1$ elements and the key of n^{th} entry of R_3 as last n element in a set to decrypt $decR_3$.
4. Arrange the upper sub-diagonal entry of matrix received as last $n-1$ elements and the key of 1^{st} entry of Q_2 as first element in a set to decrypt $decQ_2$.
5. Multiplying each element of $decQ_3$ by 10 and add it to the corresponding element of the symmetric key $decR_3$. Let us denote the resulting set as $decA_3$.
6. Find the alphabet corresponding to each value of $decA_3$ from the Table 3.
7. Find the values corresponding to each alphabet from the Table 1 to get $decR_2$.
8. Multiply each component of the symmetric key Q_2 by 26 and add in the previous set which is $decR_2$ and then subtract 1 from each element to decrypt $decA_2$.
9. Divide each element of $decA_2$ by 10 to decrypt the sets $decQ_1$ and $decR_1$ which are the sets of quotients and remainders, respectively.
10. Multiply each element of $decR_1$ by 10 and then add them to the corresponding element of $decQ_1$ to decrypt $decA_1$.
11. Find the alphabet corresponding to each element of $decA_1$ from Table 3.
12. The message is decrypted.

 doi:10.20944/preprints202410.2503.v1

Algorithm 3: Scheme 3

Encryption
INPUT : $plaintext \leftarrow$ string as received from user
OUTPUT $cipher \leftarrow$ matrix of integers
:
$A_1 \leftarrow$ Table 3 equivalents of $plaintext$
$n \leftarrow length(plaintext)$
FOR i *from* 1 *to* n **DO**
  $Q_1[i] \leftarrow quotient(A_1[i], 10)$: $R_1[i] \leftarrow remainder(A_1[i], 10)$ $A_2[i] \leftarrow 10 \cdot R_1[i] + Q_1[i]$
  $Q_2[i] \leftarrow quotient(A_2[i], 26)$ $R_2[i] \leftarrow remainder(A_2[i], 26) + 1$
END
$cypher_1 \leftarrow$ alphabets from R_2 according to Table 1
$A_3 \leftarrow$ integer equivalents of $cypher1$ as per Table 3
FOR i *from* 1 *to* n **DO**
  $Q_3[i] \leftarrow quotient(A_3[i], 10)$: $R_3[i] \leftarrow remainder(A_3[i], 10)$
  $A_4[i] \leftarrow 10 \cdot R_3[i] + Q_3[i]$: $A_5[i] \leftarrow A_3[i] + A_4[i]$
  $Q_4[i] \leftarrow quotient(A_5[i], n)$: $R_4[i] \leftarrow remainder(A_5[i], n)$
END
FOR i *from 1 to* n **DO**
  FOR j *from 1 to* n **DO**
    IF $i = j + 1$ **THEN**
      $M[i, j] \leftarrow 0 \cdot R_3[j]$
    ELSE IF $i = j - 1$ **THEN**
      $M[i, j] \leftarrow Q_2[j]$
    ELSE IF $i = j$ **THEN**
      $M[i, j] \leftarrow A_6[i]$
    ELSE
      $M[i, j] \leftarrow 0$
  END
END
$cipher \leftarrow M$
RETURN $cipher$

Decryption
INPUT : $cipher$ :: list of integers, $R_3[n], Q_2[1], n$
OUTPUT $plaintext \leftarrow$ string as recovered
:
$decR_1 \leftarrow$list comprising lower diagonal of tridiagonal matrix $cipher$
$decR_3 \leftarrow decR_1.append(R_3[n])$
$decQ_4 \leftarrow$list comprising diagonal of matrix $cipher$
$decQ_2 \leftarrow [Q_2[1],$ list comprising upper diagonal of matrix $cipher\ ]$
FOR i *from 1 to* n **DO**
  $decA_5[i] \leftarrow n \cdot decQ_4[i] + R_4[i]$: $decA_6[i] \leftarrow quotient(decA_5[i], 11)$
  $decQ_3(i) \leftarrow decA_6(i) - decR_3(i)$: $decA_3 \leftarrow 10 \cdot decQ_3(i) + decR_3(i)$
  $decCipher1 \leftarrow decCipher1.append(pytha2a(decA_3(i)))$
  $decR_2(i) \leftarrow a2pos(decCipher1(i)) - 1$
  $decA_2(i) \leftarrow 26 \cdot decQ_2(i) + decR_2(i)$
  $decQ_1(i) \leftarrow quotient(decA_2(i), 10)$: $decR_1(i) \leftarrow remainder(decA_2(i), 10)$
  $decA_1(i) \leftarrow 10 \cdot decR_1(i) + decQ_1(i)$
END
$recoveredtext \leftarrow Table3$ equivalents of $decA_1$
RETURN $recoveredtext$

 doi:10.20944/preprints202410.2503.v1

Example 3. *Plain text* $P =$ ***MATRIX*** *gives* $A_1 = [42, 11, 23, 92, 91, 63]$, $Q_1 = [4, 1, 2, 9, 9, 6]$ *and* $R_1 = [2, 1, 3, 2, 1, 3]$. *Construct a new set* $A_2 = [24, 11, 32, 29, 19, 36]$ *from the set* A_1 *by interchanging the values and the corresponding sets* $Q_2 = [0, 0, 1, 1, 0, 1]$ *and* $R_2 = [24, 11, 6, 3, 19, 10]$ *by dividing* A_2 *by* 26. *Using Table 1 on* R_2 *gives* $Cipher1 : YLGDTK$. *Using Table 3, on* $cipher1$ *obtain* $A_3 = [73, 32, 71, 41, 23, 22]$, $Q_3 = [7, 3, 7, 4, 2, 2]$ *and* $R_3 = [3, 2, 1, 1, 3, 2]$. *multiplying* R_3 *by* 10 *and then adding to the corresponding component of* Q_3 *yields* $A_4 = [37, 23, 17, 14, 32, 22]$. $A_5 = A_3 + A_4 = [110, 55, 88, 55, 55, 44]$ *and thus* $Q_4 = [18, 9, 14, 9, 9, 7]$, $R_4 = [2, 1, 4, 1, 1, 2]$ *make the cipher as:*

$$M = \begin{pmatrix} 18 & 0 & 0 & 0 & 0 & 0 \\ 3 & 9 & 1 & 0 & 0 & 0 \\ 0 & 2 & 14 & 1 & 0 & 0 \\ 0 & 0 & 1 & 9 & 0 & 0 \\ 0 & 0 & 0 & 1 & 9 & 1 \\ 0 & 0 & 0 & 0 & 3 & 7 \end{pmatrix}$$

For decryption, receive the cipher M *together with symmetric keys* Q_3 *and* n^{th} *and* 1^{st} *element of* R_3 *and* Q_2 *respectively.* $Q_3 = \{7, 3, 7, 4, 2, 2\}$, 6^{th} *element of* R_3 *is* 2 *and* 1^{st} *element of* Q_2 *is* 0. *Here* $n = 6$.
Arrange the diagonal entry in a list to decrypt $decQ_4$. *Arrange the lower sub-diagonal entry of matrix* M *as first* 5 *elements and the key of* 6^{th} *entry of* R_3 *which is* 2 *as* 6 *element in a set to decrypt* R_3. *Arrange the upper sub-diagonal entry of matrix* M *as last* 5 *elements and the key of* 1^{st} *entry of* Q_2 *which is* 0 *as* 6 *element in a list to decrypt* Q_2. $decQ_4 = [18, 9, 14, 9, 9, 7]$, $decR_3 = [3, 2, 1, 1, 3, 2]$ *and* $decQ_2 = [0, 0, 1, 1, 0, 1]$ *Multiplying the symmetric key* Q_3 *by* 10 *and adding to the corresponding element of* $decR_3$ *to decrypt the* $decA_3 = [73, 32, 71, 41, 23, 22]$.*Which on using Table 3 gives* $[Y, L, G, D, , T, K]$ *and in turn use of Table 1 yields* $decR_2 = [24, 11, 6, 3, 19, 10]$. *Multiply each element of* $decQ_2$ *by* 26 *and add in the previous set which is* $decR_2$ *and then subtract* 1 *from each element to obtain the set* $decA_2 = [24, 11, 32, 29, 19, 36]$ *and then divide each element of* $decA_2$ *by* 10 *to* $decQ_1 = [2, 1, 3, 2, 1, 3]$ *and* $decR_1 = [4, 1, 2, 9, 9, 6]$. *Multiply each element of* $decR_1$ *by* 10 *and then add them to the corresponding element of* $decQ_1$ *to get the* $decA_1 = [42, 11, 23, 92, 91, 63]$. *Find the alphabets corresponding to each element of* $decA_1$ *from the Table 3.*
$42, 11, 23, 92, 91, 63 \implies MATRIX$, *and the original message is decrypted.*

3.5. Scheme 4: Encryption and Decryption Using Tridiagonal Matrices and Disguised Key

The encryption algorithm of Scheme 4 begins by mapping plain text characters to their corresponding numerical values. These values are partitioned, transformed, and rearranged through multiple steps involving division by constants, multiplication, and addition operations. The resulting values are used to construct a tridiagonal matrix, which serves as the cipher text. Decryption, on the other hand, involves reversing these operations to accurately reconstruct the original message. The use of tridiagonal matrices and symmetric keys not only enhances the encryption process but also ensures that decryption is precise and reliable. This scheme stands out for its methodological rigor and the robust security it provides.

Scheme 4 : Description

Encryption

1. Let P be the plain text of length n.
2. Draw the Encryption Table 3.
3. Let A_1 be the set of values corresponding to the alphabets.
4. Partition the values of A_1 in two parts such that Q_1 is the set of values in A_1 that are at ten's place and R_1 is the set of values in A_1 that are at unit's place.
5. Construct a new set A_2 from the set A_1 by interchanging the values.
6. Divide each element of A_2 by 26 to get the sets Q_2 and R_2 that are the sets of quotient and the remainders respectively.

 doi:10.20944/preprints202410.2503.v1

7. Find the alphabets from the Table 1 corresponding to each element of the set R_2. Let the plain text be $cipher1$.
8. From Table 3, find the values corresponding to each alphabets of $cipher1$ and let A_3 be the ordered set containing all the values.
9. Divide each element of A_3 by 10 and let Q_3 and A_3 are the sets of quotient and remainder, respectively.
10. Multiplying R_3 by 10 and then add to the corresponding component of Q_3. The resulting values will be stored in an ordered set A_4.
11. Add the corresponding element of A_3 and A_4 and denote the resulting set by A_5.
12. Divide each element of A_5 by 11 to get the set A_6 that is the set of quotient.
13. Construct the cipher as follows:
    (a) The main diagonal will be the set A_6.
    (b) The lower sub-diagonal set as entries from the first $n-1$ elements of the set R_3.
    (c) The entries of the upper sub-diagonal are the last $n-1$ elements of the set Q_2.
14. We obtained the cipher.

Decryption

1. Let the receiver has received the message as a matrix of order $n \times n$ together with two symmetric keys of n^{th} and 1^{st} entries of R_3 and Q_2, respectively.
2. Arrange the diagonal entry in a set to decrypt $decA_6$.
3. Arrange the lower sub-diagonal entry of matrix received as first $n-1$ elements and the key of n^{th} entry of R_3 as last n element in a set to decrypt $decR_3$.
4. Arrange the upper sub-diagonal entry of matrix received as last $n-1$ elements and the key of 1^{st} entry of Q_2 as first element in a set to decrypt $decQ_2$.
5. Subtract each element of $decR_3$ from the corresponding elements of $decA_6$ and the resulting set will be denoted by $decQ_3$.
6. Find $decA_3$ by multiplying each element of $decQ_3$ by 10 and then add it to the corresponding element of $decR_3$.
7. Find the alphabets corresponding to each value of $decA_3$ from the Table 3.
8. Find the values corresponding to each alphabet from the Table 1 to get $decR_2$.
9. Multiply each element of $decQ_2$ by 26 and add in the previous set $decR_2$ to decrypt $decA_2$.
10. Divide each element of $decA_2$ by 10 to decrypt the sets $decQ_1$ and $decR_1$ which are the sets of quotients and remainders, respectively.
11. Multiply each element of $decR_1$ by 10 and then add them to the corresponding elements of $decQ_1$ to decrypt $decA_1$.
12. Find the alphabets corresponding to each element of $decA_1$ from Table 3.
13. Original plain text is decrypted.

Example 4. *Plaintext* $P =$ ***MATRIX*** *gives* $A_1 = [42, 11, 23, 92, 91, 63]$, $Q_1 = [4, 1, 2, 9, 9, 6]$ *and* $R_1 = [2, 1, 3, 2, 1, 3]$. *Interchanging the values of* A_1 *gives* $A_2 = [24, 11, 32, 29, 19, 36]$, *and also by dividing 26 we have* $Q_2 = \{0, 0, 1, 1, 0, 1\}$ *and* $R_2 = [24, 11, 6, 3, 19, 10]$. *Using Table 1 on* R_2 *gives* $Cipher1 : YLGDTK$. *Table 3, on* $cipher1$ *yields* $A_3 = [73, 32, 71, 41, 23, 22]$, $Q_3 = [7, 3, 7, 4, 2, 2]$ *and* $R_3 = [3, 2, 1, 1, 3, 2]$. *Multiplying* R_3 *by 10 and then add to the corresponding element of* Q_3 *to obtain* $A_4 = [37, 23, 17, 14, 32, 22]$. *Next* $A_3 + A_4 = A_5 = [110, 55, 88, 55, 55, 44]$ *divided by 11,* $A_6 = [10, 5, 8, 5, 5, 4]$. *This allows construction of cipher as*

$$M = \begin{pmatrix} 10 & 0 & 0 & 0 & 0 & 0 \\ 3 & 5 & 1 & 0 & 0 & 0 \\ 0 & 2 & 8 & 1 & 0 & 0 \\ 0 & 0 & 1 & 5 & 0 & 0 \\ 0 & 0 & 0 & 1 & 5 & 1 \\ 0 & 0 & 0 & 0 & 3 & 4 \end{pmatrix}$$

 doi:10.20944/preprints202410.2503.v1

Algorithm 4: Scheme 4

Encryption
INPUT : $plaintext \leftarrow$ string as received from user
OUTPUT $cipher \leftarrow$ matrix of integers
:
$A_1 \leftarrow$ Table 3 equivalents of $plaintext$
$n \leftarrow length(plaintext)$
FOR i *from* 1 *to* n **DO**
  $Q_1[i] \leftarrow quotient(A_1[i], 10)$: $R_1[i] \leftarrow remainder(A_1[i], 10)$ $A_2[i] \leftarrow 10 \cdot R_1[i] + Q_1[i]$
  $Q_2[i] \leftarrow quotient(A_2[i], 26)$ $R_2[i] \leftarrow remainder(A_2[i], 26) + 1$
END
$cypher_1 \leftarrow$ alphabets from R_2 according to Table 1
$A_3 \leftarrow$ integer equivalents of $cypher1$ as per Table 3
FOR i *from* 1 *to* n **DO**
  $Q_3[i] \leftarrow quotient(A_3[i], 10)$: $R_3[i] \leftarrow remainder(A_3[i], 10)$
  $A_4[i] \leftarrow 10 \cdot R_3[i] + Q_3[i]$: $A_5[i] \leftarrow A_3[i] + A_4[i]$
  $Q_4[i] \leftarrow quotient(A_5[i], n)$: $R_4[i] \leftarrow remainder(A_5[i], n)$
END
FOR i *from 1 to* n **DO**
  FOR j *from 1 to* n **DO**
    IF $i = j + 1$ **THEN**
      $M[i, j] \leftarrow 0 \cdot R_3[j]$
    ELSE IF $i = j - 1$ **THEN**
      $M[i, j] \leftarrow Q_2[j]$
    ELSE IF $i = j$ **THEN**
      $M[i, j] \leftarrow A_6[i]$
    ELSE
      $M[i, j] \leftarrow 0$
  END
END
$cipher \leftarrow M$
RETURN $cypher$

Decryption
INPUT : $cypher$ :: list of integers, $R_3[n], Q_2[1], n$
OUTPUT $recoveredtext \leftarrow$ string as recovered
:
$decR_3 \leftarrow [lowerDiagonal(cipher), R_3[3]]$
$decA_6 \leftarrow [diagonal(cipher)]$
$decQ_2 \leftarrow [Q_2[1], upperDiagonal(cipher)]$
$decQ_3 \leftarrow decA_6 - decR_3$
$decA_3 \leftarrow 10 \cdot decQ_3 + decR_3$
$decCipher1 \leftarrow$ Table 3 equivalents of $decA_3$
FOR i *from 1 to* n **DO**
  $decR_2(i) \leftarrow$ Table $1(decCipher1(i)) - 1$
  $decA_2(i) \leftarrow 26 \cdot decQ_2(i) + decR_2(i)$
  $decQ_1(i) \leftarrow quotient(decA_2(i), 10)$: $decR_1(i) \leftarrow remainder(decA_2(i), 10)$
  $decA_1(i) \leftarrow 10 \cdot decR_1(i) + decQ_1(i)$
END
RETURN Table $3(decA_1)$
RETURN $plaintext$

 doi:10.20944/preprints202410.2503.v1

Table 4. Complexity Analysis: Encryption in S_2 is of linear order, while all others due to involvement of square matrices have square order complexities.

	Time		Space	
	Encryption	Decryption	Encryption	Decryption
S_1	n^2	n^2	n^2	n^2
S_2	n	n^2	n^2	n^2
S_3	n^2	n^2	n^2	n^2
S_4	n^2	n^2	n^2	n^2

For decryption, on receiving the cipher together with symmetric keys: 6^{th} and 1^{st} element of R_3 and Q_2, respectively. Using 6^{th} element of R_3 is 2 and 1^{st} element of Q_2 is 0. Here, $n = 6$. Arrange the diagonal entry in a set to decrypt A_6. Arrange the upper sub-diagonal entry of matrix M as last 5 elements and the key of 1^{st} entry of Q_2 which is 0 as 6 element in a set to decrypt Q_2. We now have $decA_6 = [10, 5, 8, 5, 5, 4]$, $decR_3 = [3, 2, 1, 1, 3, 2]$ and $decQ_2 = [0, 0, 1, 1, 0, 1]$. Subtracting $decR_3$ from $decA_6$, $decQ_3 = [7, 4, 7, 4, 2, 2]$. Multiplying $decQ_3$ by 10 and adding to the corresponding element of $decR_3$, gives $decA_3 = [73, 42, 71, 41, 23, 22]$ and use of Table 3 $\implies$ Y, L, G, D, T, K. Encoding this with Table 1 produces $decR_2 = [24, 11, 6, 3, 19, 10]$. Now multiply each element of $decQ_2$ by 26 and add in the previous set which is $decR_2$ to obtained the set $decA_2 = [24, 11, 32, 29, 19, 36]$, $decQ_1 = \{2, 1, 3, 2, 1, 3$ and $decR_1 = [4, 1, 2, 9, 9, 6]$. Multiply each element of $decR_1$ by 10 and then add them to the corresponding element of $decQ_1$ to get the set $decA_1 = [42, 11, 23, 92, 91, 63]$. Table 3 gives $42, 11, 23, 92, 91, 63 \implies MATRIX$, and the message is decrypted.

3.6. Complexity Analysis

We only give details of time-complexity and space-complexity for Scheme 1. **Encryption:** Assigning the length n to the plaintext P and drawing the encryption table both take $O(1)$ time. Extracting the set A_1, partitioning A_1 into Q_1 and R_1, constructing A_2, partitioning A_2 into Q_2 and R_2, and constructing the diagonal matrix M_2 all involve $O(n)$ operations. Constructing the matrix M_1, adding M_1 and M_2, and constructing the final cipher list require $O(n^2)$ time. Hence the total time complexity for the encryption process is:

$$O(n) + O(n) + O(n) + O(n^2) + O(n) + O(n^2) + O(n^2) = O(n^2)$$

Decryption: Receiving the list takes $O(1)$ time, while counting the elements in the cipher, taking the square root, and constructing the matrix $decM$ each require $O(n^2)$. Arranging diagonal entries into an array, constructing $decR_2$, subtracting arrays to obtain $decQ_2$, multiplying and adding elements to form $decA_2$, dividing to get $decQ_1$ and $decR_1$, reconstructing $decA_1$, and finding corresponding alphabets all involve $O(n)$ operations. Therefore the total time complexity for the decryption process is:

$$O(1) + O(n^2) + O(n^2) + O(n) + O(n) + O(n) + O(n) + O(n) + O(n) + O(n) = O(n^2)$$

Hence, the overall time complexity of Scheme 1 is:

$$\boxed{O(n^2)}$$

All other Schemes may be analysed in a similar fashion. Complete data, thus obtained is shown in Table 4.

4. Conclusion

We have created four schemes. The first encrypts data using a string of numbers. n^2 numbers make up a cipher, where n is the length of the plain-text. The encryption/decryption algorithm does not have a key and does not provide an attacker with a pattern to access the original message. The length of the plain text can be predicted by the attacker from the cypher-text, which is the only

 doi:10.20944/preprints202410.2503.v1

disadvantage.

An additional layer has been utilised in Scheme 2. Two keys are required in order to decrypt the information. Before being transformed into a list of n^2 numbers, plain-text is first disguised as another plain-text of the same length. Furthermore, in order to decrypt the original data, two keys, each made up of n numbers, are required. We'll improve our data security with this plan. A word of length n can only be decrypted by the receiver if he had $n^2 + 2n$ numbers, which is the only drawback.

Scheme three has two layers, just as Scheme 2. In this scheme, the cipher consists of an ordered set of $2n - 2$ numbers and three keys, each of size n. We make it difficult for attackers to predict the length of the plain-text by implementing this scheme. We also shortened the length of the keys and the cipher-text. In scheme 4, we have not only shrunk the key's size but also changed its ordering, making it harder for an attacker to find the original text.

References

1. M. Lalitha.; S. Vasu. A Study On Graph Theory In Cryptography Using Python. *Journal of Engineering Technologies and Innovative Research* **2023**, *10*.
2. Amudha, P.; Sagayaraj, A.C.C.; Sheela, A.C.S. An Application of Graph Theory in Cryptography. *International Journal of Pure and Applied Mathematics* **2018**, *119*, 375–383.
3. Chowdhury, S.; Ghosh, P.; Jana, M. An Approach of Graph Theory for Solving Cryptographic Problem. *BKGC Scholars* **2020**, *1*, 64–68.
4. Kumari, M.; Kirubanad, V. Data Encryption And Decryption Using Graph Plotting. *International Journal of Civil Engineering and Technology* **2018**, *9*, 36–46.
5. Meenakshi, A.; Kannan, A.; Cep, R.; Elangovan, M. Efficient Graph Network Using Total Magic Labeling and Its Applications. *Mathematics* **2023**, *11*, 4132. doi:10.3390/math11194132.
6. G. S . Wijesiri, P.A.S.D.P. Encryption and Decryption Algorithms in Symmetric Key Cryptography Using Graph Theory. *Psychology and Education Journal* **2021**, *58*, 3420–3427. doi:10.17762/pae.v58i1.1280.
7. Mahboob, A.; Asif, M.; Nadeem, M.; Saleem, A.; Eldin, S.M.; Siddique, I. A Cryptographic Scheme for Construction of Substitution Boxes Using Quantic Fractional Transformation. *IEEE Access* **2022**, *10*, 132908–132916. doi:10.1109/ACCESS.2022.3230141.
8. Godquin, T.; Barbier, M.; Gaber, C.; Grimault, J.L.; Le Bars, J.M. Applied graph theory to security: A qualitative placement of security solutions within IoT networks. *Journal of Information Security and Applications* **2020**, *55*, 102640. doi:10.1016/j.jisa.2020.102640.
9. Yang, R.; Yu, A.; Cai, L.; Meng, D. Subspace clustering via graph auto-encoder network for unknown encrypted traffic recognition. *Cybersecurity* **2022**, *5*, 29. doi:10.1186/s42400-022-00131-y.
10. Zhang, X.; Zhang, S.; Ye, C.; Yao, B. Graphic lattices made by graph felicitous-type labelings and colorings of topological coding. *Discrete Applied Mathematics* **2023**, *336*, 37–46. doi:10.1016/j.dam.2023.03.023.
11. Yang, Z.; Yu, L.; Liu, Y.; Alotaibi, N.D.; Alsaadi, F.E. Event-triggered privacy-preserving bipartite consensus for multi-agent systems based on encryption. *Neurocomputing* **2022**, *503*, 162–172. https://doi.org/10.1016/j.neucom.2022.06.074.