



A Cryptographic Scheme Involving Baudhāyana Triplets, XOR Logic and Decomposition of a Matrix

Deepali Chouhan¹, S. S. Shrivastava²

^{1,2}Department of Mathematics Institute for Excellence in Higher Education, Bhopal

ARTICLE INFO	ABSTRACT
Published Online: 02 March 2026	The objective of this paper is to develop an algorithm for the encryption and decryption of information using Baudhāyana triplets, XOR logic and LU decomposition-based key generation from a non-singular matrix. A detailed numerical example is provided to demonstrate correctness, reversibility, and feasibility. This work establishes a meaningful bridge between ancient Indian mathematical knowledge systems (IKS) and modern lightweight cryptography.
Corresponding Author: Deepali Chouhan	
KEYWORDS: Śulbasūtra, Baudhāyana Triplets, XOR Logic, LU decomposition of a Matrix, Cryptography	

1. INTRODUCTION

1.1 Cryptography:

In the digital era, protecting information from unauthorized access has become a major challenge, making data security an essential requirement. Cryptography [3, 5, 8] is the most commonly used approach for ensuring the security of information, particularly during data storage and message transmission. By preventing unauthorized access to sensitive data, cryptography ensures confidentiality and privacy. The transformation of meaningful information into an unreadable format is known as encryption, whereas the process of converting this unreadable data back into its original form is called decryption. The original readable message is referred to as plaintext, while its encoded version is known as ciphertext. Although ciphertext appears meaningless, it fully represents the information contained in the plaintext. Without proper decryption, neither humans nor machines can recover the original message. Cryptographic systems operate using auxiliary parameters known as keys, and variations in these keys alter the encryption process, thereby modifying the internal functioning and enhancing the overall security of the algorithm.

1.2 Baudhāyana Triplets:

Baudhāyana's Śulbasūtra represents one of the earliest systematic treatments of geometry and number theory in human history. Among its most notable contributions is the formulation of integer triplets satisfying the relation $a^2 + b^2 = c^2$, now known as Baudhāyana or Pythagorean triplets. These triplets exhibit strong numerical structure, cyclic

generation, and deterministic relationships, properties that are highly desirable in cryptographic key construction.

Baudhāyana triplets [1, 9] are sets of three positive integers (a, b, c) that satisfy the relation:

$$a^2 + b^2 = c^2$$

This relationship is explicitly stated in the Baudhāyana Śulbasūtra (c. 800–600 BCE), making it one of the earliest known formulations of the theorem later attributed in Europe to Pythagoras. These triplets were originally developed for accurate geometric constructions, particularly in the construction of Vedic fire altars (vedīs).

Long before modern cryptography, Baudhāyana (c. 800–600 BCE) introduced structured numerical relationships in the *Baudhāyana Śulbasūtra*, especially through what is now called the Pythagorean theorem:

$$a^2 + b^2 = c^2$$

The Baudhāyana triplets generated from this relation are deterministic, scalable, and cyclic, qualities that make them excellent candidates for cryptographic key material.

Common examples include:

a	b	c
3	4	5
5	12	13
8	15	17
7	24	25

A general formula for generating triplets is:

$$a = m^2 - n^2, b = 2mn, c = m^2 + n^2$$

where $m > n$, $m, n \in \mathbb{Z}^+$.

Baudhāyana triplets exhibit important numerical properties such as determinism, scalability, and balance, which make them relevant even in modern contexts such as

algorithm design and cryptography. Their continued study highlights the depth of ancient Indian mathematical knowledge and its enduring relevance to contemporary scientific and technological applications.

1.3 XOR Logic:

XOR (Exclusive OR) logic [7, 4] plays a fundamental role in modern cryptography due to its simplicity, efficiency, and reversible nature. Operating at the binary level, XOR compares two input bits and produces an output of 1 when the inputs differ and 0 when they are the identical. This logical behavior makes XOR especially suitable for encryption, as a message combined with a key using XOR can be reversed by applying the same operation again with the same key. As a result, XOR forms the core of many symmetric encryption schemes, stream ciphers, and key-mixing processes. In cryptographic systems, XOR is widely used to blend plaintext with secret keys, introduce diffusion, and obscure statistical patterns present in the original data. Its low computational cost enables fast execution, making it ideal for real-time applications, embedded systems, and lightweight security protocols. Although XOR alone does not provide strong security, its integration with structured key generation, cyclic transformations, and additional mathematical operations significantly enhances cryptographic strength. Consequently, XOR remains an essential building block in both classical and modern cryptographic algorithms.

XOR (Exclusive OR) operation:

A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

Key property:

$$C = M \oplus K \Rightarrow M = C \oplus K$$

This perfect reversibility makes XOR ideal for symmetric encryption.

1.4 LU Decomposition:

In the LU decomposition technique [2, 6], a matrix can be represented as the product of a lower and an upper triangular matrix, provided that all its leading principal minors are non-zero.

Let B be a non-singular matrix of order n. Then B can be expressed as the product of two triangular matrices, denoted by X and Y, where X is a lower triangular matrix and Y is an upper triangular matrix. Thus,

$$B = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \dots & \dots & \dots & \dots \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{bmatrix} = \begin{bmatrix} x_{11} & 0 & \dots & 0 \\ x_{21} & x_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nn} \end{bmatrix} \begin{bmatrix} y_{11} & y_{12} & \dots & y_{1n} \\ 0 & y_{22} & \dots & y_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & y_{nn} \end{bmatrix}$$

$$\text{where } X = \begin{bmatrix} x_{11} & 0 & \dots & 0 \\ x_{21} & x_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nn} \end{bmatrix} \quad \text{and} \quad Y = \begin{bmatrix} y_{11} & y_{12} & \dots & y_{1n} \\ 0 & y_{22} & \dots & y_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & y_{nn} \end{bmatrix}$$

There are three types of decomposition methods, namely Doolittle, Crout and Cholesky. In this paper, we used Crout's method. In this method, we choose $(y_{11}, y_{22}, \dots, y_{nn}) = (1, 1, \dots, 1)$, therefore

$$\begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \dots & \dots & \dots & \dots \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{bmatrix} = \begin{bmatrix} x_{11} & 0 & \dots & 0 \\ x_{21} & x_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nn} \end{bmatrix} \begin{bmatrix} 1 & y_{12} & \dots & y_{1n} \\ 0 & 1 & \dots & y_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

$$\text{where } X = \begin{bmatrix} x_{11} & 0 & \dots & 0 \\ x_{21} & x_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nn} \end{bmatrix} \quad \text{and} \quad Y = \begin{bmatrix} 1 & y_{12} & \dots & y_{1n} \\ 0 & 1 & \dots & y_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

2. LITERATURE REVIEW

Adigun [1], Yegnanarayanan [9], Samriddhi [7], Garg [4], Dixit [2], Mittal [6] and other researchers have proposed various algorithms based on Baudhāyana Triplets, XOR (Exclusive OR) logic and LU decomposition independently. For encrypting the plaintext, some researchers applied the Baudhāyana Triplets and XOR logic, and for decrypting the ciphertext, they apply the corresponding inverse logic. On the other hand, in the LU decomposition technique, some researchers used the lower triangular matrix as the encryption key and the upper triangular matrix as decryption key, under computations performed modulo a prime number.

3. METHODOLOGY

Following the work of Adigun [1], Yegnanarayanan [9], Samriddhi [7], Garg [4], Dixit [2], Mittal [6], and other researchers, we introduce a new algorithm involving Baudhāyana Triplets, XOR (Exclusive OR) logic, along with the LU decomposition technique. Thus, we propose a multiple encryption system. In this technique, we first apply the Baudhāyana Triplets and XOR (Exclusive OR) logic as first encryption key and generate an intermediate ciphertext. Then, we apply the lower triangular matrix (generated from key matrix) as the second encryption key to obtain the final ciphertext. In the reverse process, we use the upper triangular matrix (generated from key matrix) as the first decryption key to obtain the intermediate plaintext. Finally, Baudhāyana triplets and XOR (Exclusive OR) logic are applied as the second decryption key to retrieve the original plaintext.

In this technique, we use a Baudhāyana triplet (a, b, c) and cyclic key variation $K_i = (a_i \oplus b_i \oplus c_i) \bmod 256$, which generate the first encryption key. Further, using the LU Decomposition method, a matrix A is expressed as $A = LU$, subject to the condition $\text{g.c.d}((\det A) \bmod q, q) = 1$, to ensure the invertibility of the key matrix. The constant matrix is then compute using $\text{Cons} = (\text{key matrix } A) P(\bmod p)$, where $p = 256$ (choose). We compute $C_i = L^{-1} \text{Cons}$ for encryption process and $M = U^{-1} C_i$ for decryption process, where M denotes the plaintext and C_i denotes ciphertext.

The integration of Baudhāyana triplets, XOR (Exclusive OR) logic, and a secret key enhances the secrecy of the system. Without the appropriate key and the specific LU Decomposition of the matrix, breaking the ciphertext becomes computationally difficult. Moreover, increasing the number of elements in the sequence strengthens resistance against cryptographic attacks.

4. KEY GENERATION

4.1 Key Generation Using Baudhāyana Triplets and XOR Logic:

Select a Baudhāyana triplet (a, b, c). Define the encryption key as:

$$K = (a \oplus b \oplus c)$$

To strengthen security, **cyclic key variation** can be used:

$$K_i = (a_i \oplus b_i \oplus c_i) \bmod 256$$

where successive triplets are generated by incrementing m, n.

(Table 1)

Triplets (For Odd Numbers)					ASCII Code in Binary			$X_i = a_i \oplus b_i \oplus c_i$	Decimal Value (Say D_i)
i	a _i	b _i	c _i						
1	3	4	5		0000 0011	0000 0100	0000 0101	0000 0010	2
2	5	1 2	1 3		0000 0101	0000 1100	0000 1101	0000 0100	4
3	7	2 4	2 5		0000 0111	0001 1000	0001 1001	0000 0110	6
4	9	4 0	4 1		0000 1001	0010 1000	0010 1001	0000 1000	8
5	1 1	6 0	6 1		0000 1011	0011 1100	0011 1101	0000 1010	10
6	1 3	8 4	8 5		0000 1101	0101 0100	0101 0101	0000 1100	12
7	1 5	1 1 2	1 1 3		0000 1111	0111 0000	0111 0001	0000 1110	14
8	1 7	1 4 4	1 4 5		0001 0001	1001 0000	1001 0001	0001 0000	16

9	1	1	1	0001	1011	1011	0001	18
	9	8	8	0011	0100	0101	0010	
		0	1					

(Table 2)

i	D_i	$D_i (\bmod 256)$	$K_i = D_i (\bmod 256)$ (in binary form)
1	2	2	00000010
2	4	4	00000100
3	6	6	00000110
4	8	8	00001000
5	10	10	00001010
6	12	12	00001100
7	14	14	00001110
8	16	16	00010000
9	18	18	00010010

4.2 Key Generation Using LU Decomposition of a Matrix:

Consider a non-singular square matrix of order 3, randomly chosen and denoted by A, as an intermediate key matrix, given as follows:

$$A = \begin{bmatrix} 5 & 10 & 10 \\ 4 & 9 & 13 \\ 1 & 4 & 15 \end{bmatrix}$$

Choose $A = LU$

where

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \quad L = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \quad \text{and}$$

$$U = \begin{bmatrix} 1 & u_{12} & u_{13} \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

In Crout's method, consider $(u_{11}, u_{22}, u_{33}) = (1, 1, 1)$ for LU decomposition of matrix A, therefore

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} 1 & u_{12} & u_{13} \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

Hence

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} l_{11} & l_{11}u_{12} & l_{11}u_{13} \\ l_{21} & l_{21}u_{12} + l_{22} & l_{21}u_{13} + l_{22}u_{23} \\ l_{31} & l_{31}u_{12} + l_{32} & l_{31}u_{13} + l_{32}u_{23} + l_{33} \end{bmatrix}$$

Equate the corresponding elements both sides, we get

$$a_{11} = l_{11}, \quad a_{21} = l_{21}, \quad a_{31} = l_{31}$$

$$a_{12} = l_{11}u_{12}, \quad a_{22} = l_{21}u_{12} + l_{22}, \quad a_{32} = l_{31}u_{12} + l_{32}$$

$$a_{13} = l_{11}u_{13}, \quad a_{23} = l_{21}u_{13} + l_{22}u_{23}, \quad a_{33} = l_{31}u_{13} + l_{32}u_{23} + l_{33}$$

Here

$$a_{11} = l_{11} = 5, \quad a_{21} = l_{21} = 4, \quad a_{31} = l_{31} = 1,$$

$$u_{11} = u_{22} = u_{33} = 1$$

$$a_{12} = 10, a_{22} = 9, a_{32} = 4, a_{13} = 10, \\ a_{23} = 13, a_{33} = 15,$$

$$a_{12} = l_{11}u_{12} \Rightarrow u_{12} = \frac{a_{12}}{l_{11}} = \frac{10}{5} = 2$$

$$a_{13} = l_{11}u_{13} \Rightarrow u_{13} = \frac{a_{13}}{l_{11}} = \frac{10}{5} = 2$$

$$a_{22} = l_{21}u_{12} + l_{22} \Rightarrow l_{22} = a_{22} - l_{21}u_{12} = 9 - 4 \times 2 = 1$$

$$a_{32} = l_{31}u_{12} + l_{32} \Rightarrow l_{32} = a_{32} - l_{31}u_{12} = 4 - 1 \times 2 = 2$$

$$a_{23} = l_{21}u_{13} + l_{22}u_{23} \Rightarrow u_{23} = \frac{a_{23} - l_{21}u_{13}}{l_{22}} = \frac{13 - 4 \times 2}{1} = 5$$

$$a_{33} = l_{31}u_{13} + l_{32}u_{23} + l_{33}$$

$$\Rightarrow l_{33} = a_{33} - l_{31}u_{13} - l_{32}u_{23}$$

$$\Rightarrow l_{33} = 15 - 1 \times 2 - 2 \times 5 = 3$$

Therefore

$$L = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} = \begin{bmatrix} 5 & 0 & 0 \\ 4 & 1 & 0 \\ 1 & 2 & 3 \end{bmatrix}$$

and

$$U = \begin{bmatrix} 1 & u_{12} & u_{13} \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 2 \\ 0 & 1 & 5 \\ 0 & 0 & 1 \end{bmatrix}$$

5. PROPOSED ENCRYPTION-DECRYPTION MODEL (ALGORITHM)

5.1 Encryption Algorithm:

1. Consider the plaintext message.
2. Obtain the ASCII values of each character of plaintext message in decimal form and their corresponding binary representation (denoted by M_i).
3. Generate the first encryption key $K_i = D_i \pmod{256}$ (in binary form), where D_i is the decimal value of $X_i = a_i \oplus b_i \oplus c_i$ and (a, b, c are Baudhāyana triplets corresponding to odd numbers).
4. For each plaintext byte M_i , calculate $C_i = M_i \oplus K_i$.
5. Convert each $C_i = (C_1, C_2, \dots, C_n)$ into its corresponding ASCII symbol to obtain the intermediate ciphertext.
6. Arrange the decimal values of intermediate ciphertext into a square matrix of order 3×3 . Denote this matrix by P, referred to as the intermediate plaintext matrix.
7. Let A be a randomly chosen non-singular square matrix of order 3×3 , serving as the key matrix.
8. Find $A = LU$ by Crout's method.
9. Calculate the constant matrix using the following formula:
Cons (say) = (key matrix A) P (mod p),
where $p = 256$ (choose)
10. Compute the final ciphertext matrix C (say) using the formula

$$C = L^{-1} \text{Cons} \pmod{p},$$

where $p = 256$ (choose) and L is the lower triangular matrix used as the second encryption key.

11. Convert each element of matrix C into its corresponding ASCII character, row by row, using ASCII table, we get final ciphertext.

5.2 Decryption Algorithm:

1. Consider the ciphertext.
2. Convert the ciphertext into its corresponding decimal values using ASCII table and arrange them in matrix form, denoted by I_c , row by row.
3. Obtain the intermediate plaintext matrix P (say) using the formula

$$P = U^{-1}I_c \pmod{256},$$

where U is the upper triangular matrix generated from the key matrix using Court's LU decomposition and is used as the decryption key.

4. Convert each element of matrix P into its corresponding ASCII character, row by row, to obtain the intermediate plaintext.
5. Obtain the final plaintext in binary form using the formula $M_i = P_i \oplus K_i$.
6. Convert each M_i into its corresponding ASCII character to recover the original plaintext message.

6. ILLUSTRATION

6.1 Encryption:

1. Consider
 - Plaintext: “BTRIPLETS”
 - ASCII values:

Message	Decimal Code	Binary Code (say M_i)
B	66	01000010
T	84	01010100
R	82	01010010
I	73	01001001
P	80	01010000
L	76	01001100
E	69	01000101
T	84	01010100
S	83	01010011

Character	M_i	K_i (Key)	$C_i = M_i \oplus K_i$	Symbol from ASCII
B	01000010	00000010	01000000	@
T	01010100	00000100	01010000	P
R	01010010	00000110	01010100	T
I	01001001	00001000	01000001	A
P	01010000	00001010	01011010	Z
L	01001100	00001100	01000000	@
E	01000101	00001110	01001011	K
T	01010100	00010000	01000100	D

S	0101001	0001001	0100000	A
	1	0	1	

2. Convert each C_i into its corresponding character using the ASCII table. We obtain the intermediate ciphertext as follows:

@PTAZ@KDA

3. Arrange the characters of the intermediate ciphertext into a square matrix of order 3×3 . Then, convert each character into its corresponding numeric value using the ASCII code. We obtain a matrix P (say), called the intermediate plaintext matrix, as follows:

$$P = \begin{bmatrix} 64 & 80 & 84 \\ 65 & 90 & 64 \\ 75 & 68 & 65 \end{bmatrix}$$

4. Calculate the constant matrix using the following formula:

$$\text{Cons}(\text{say}) = (\text{key matrix A}) P (\text{mod } p),$$

where $p = 256$ (choose)

$$\text{So Cons} = \begin{bmatrix} 5 & 10 & 10 \\ 4 & 9 & 13 \\ 1 & 4 & 15 \end{bmatrix} \begin{bmatrix} 64 & 80 & 84 \\ 65 & 90 & 64 \\ 75 & 68 & 65 \end{bmatrix} (\text{mod } 256)$$

$$= \begin{bmatrix} 1720 & 1980 & 1710 \\ 1816 & 2014 & 1757 \\ 1449 & 1460 & 1315 \end{bmatrix} (\text{mod } 256)$$

$$= \begin{bmatrix} 184 & 188 & 174 \\ 24 & 222 & 221 \\ 169 & 180 & 35 \end{bmatrix}$$

5. Now calculate the final ciphertext matrix C (say) by the formula

$$C = L^{-1} \text{Cons} (\text{mod } p),$$

where $p = 256$ (choose) and L is lower triangular matrix used as encryption key, therefore

$$C = \begin{bmatrix} 205 & 0 & 0 \\ 204 & 1 & 0 \\ 137 & 170 & 171 \end{bmatrix} \begin{bmatrix} 184 & 188 & 174 \\ 24 & 222 & 221 \\ 169 & 180 & 35 \end{bmatrix} (\text{mod } 256)$$

$$= \begin{bmatrix} 37720 & 38540 & 35670 \\ 37560 & 38574 & 35717 \\ 58187 & 94276 & 67393 \end{bmatrix} (\text{mod } 256)$$

$$= \begin{bmatrix} 88 & 140 & 86 \\ 184 & 174 & 133 \\ 75 & 68 & 65 \end{bmatrix}$$

6. Now convert each element of matrix C into its corresponding character row by row using the ASCII table to obtain the final ciphertext as follows:

XEV, ®...KDA

6.2 Decryption:

1. Consider the ciphertext and key received from the sender i.e.

XEV, ®...KDA

2. Convert the ciphertext into its corresponding decimal values using the ASCII table and arrange them in matrix form row by row, we get

$$\begin{bmatrix} 88 & 140 & 86 \\ 184 & 174 & 133 \\ 75 & 68 & 65 \end{bmatrix} = I_c (\text{say})$$

3. Now obtain the intermediate plaintext matrix P (say) using the formula

$$P = U^{-1} I_c (\text{mod } 256),$$

where U is an upper triangular matrix (generated from key matrix using Court's LU decomposition) and is used as decryption key. Therefore

$$P = \begin{bmatrix} 1 & 254 & 8 \\ 0 & 1 & 251 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 88 & 140 & 86 \\ 184 & 174 & 133 \\ 75 & 68 & 65 \end{bmatrix} (\text{mod } 256)$$

$$= \begin{bmatrix} 47424 & 44880 & 34388 \\ 19009 & 17242 & 16448 \\ 75 & 68 & 65 \end{bmatrix} (\text{mod } 256)$$

$$= \begin{bmatrix} 64 & 80 & 84 \\ 65 & 90 & 64 \\ 75 & 68 & 65 \end{bmatrix}$$

4. Now, convert each element of matrix P into its corresponding character row by row using the ASCII table, we obtain the intermediate plaintext as follows:

@PTAZ@KDA

5. To obtain the final plaintext message, we apply the following process:

P	P _i	K _i	M _i = P _i ⊕ K _i	Symbol from ASCII
@	01000000	00000010	01000010	B
P	01010000	00000100	01010100	T
T	01010100	00000110	01010010	R
A	01000001	00001000	01001001	I
Z	01011010	00001010	01010000	P
@	01000000	00001100	01001100	L
K	01001011	00001110	01000101	E
D	01000100	00010000	01010100	T
A	01000001	00010010	01010011	S

6. Therefore, the original plane text message is as follows:

"BTRIPLETS"

7. RESULT AND DISCUSSION

The widespread adoption of e-services and enhanced internet-based platforms by sectors such as banking and financial institutions has significantly transformed service delivery. However, the growing reliance on digital systems has also created new avenues for financial misconduct. Among various cybercrimes, online banking fraud has emerged as one of the most critical challenges. To address this issue, the key generation scheme proposed in this study can be effectively utilized as a fraud prevention mechanism.

In this paper, we develop a new cryptographic scheme using Baudhāyana triplets, XOR logic, and LU decomposition of a non-singular matrix. We propose a multiple encryption system that employs a double encryption key. First, Baudhāyana triplets and XOR logic are applied to obtain an intermediate ciphertext. Then, a second key derived from the lower triangular matrix is applied to produce the

final ciphertext. Consequently, it becomes computationally difficult to recover the key through cryptographic attacks.

Therefore, it is possible to extend this work further. A dynamic key theory plays an important role in reducing the risk of cryptanalytic attacks. Hence, in such situations, the method presented in this paper is useful.

8. CONCLUSION

Since the proposed algorithm is developed using Baudhāyana triplet, XOR logic, and LU decomposition of a non-singular matrix, it provides a robust encryption strategy. In this algorithm, the secret key is shared between the sender and the receiver, and the ciphertext is generated using modular arithmetic.

Due to the decomposition of the matrix into lower and upper triangular matrices, symmetric key generation becomes highly secure for both encryption and decryption of messages. In the proposed scheme, over a residue class modulo a number, the lower triangle matrix is used for encryption, while the upper triangle matrix is used for decryption. The proposed cryptosystem offers enhanced protection against plaintext attack and ciphertext attack.

Therefore, a new encryption algorithm is developed in which four levels of security are incorporated, namely the Baudhāyana triplet, XOR logic, LU decomposition, and a secret key mechanism. The encryption and decryption times can be analysed with respect to varying input sizes, and it is observed that both increase proportionally as the input size increases.

Such models not only enhance lightweight encryption techniques but also reaffirm the global scientific significance of the Baudhāyana triplet.

REFERENCES

1. Adigun A. A., Jimoh K. O., Adebayo Y. O., Kolawole M. O.: Review of Pythagorean Triple Based cryptography System for Information Security, University of Pitesti Scientific Bulletin: Electronics and Computers Science, ISSN – 2344 – 2166, Vol.21, Issue 1, 2021.
2. Dixit Sandeep, Dobahl Girish, Pandey Shweta: Encrypt and Decrypt Messages Based on LU Decomposition Using Multiple Keys, International Journal of Scientific & Technology Research, ISSN 2277-8616, Vol. 8, Issue 11, November 2019 pp. 3347-3351.
3. Forouzan Behrouz A: Cryptography & Network Security, McGraw Hill Education, 2007.
4. Garg Satish Kumar: Cryptography Using XOR Cipher, Research Journal of Science and Technology, ISSN 0975-4393 (Print), Vol. 09, Issue-01, January -March 2017.
5. Kahate Atul: Cryptography and Network Security, Tata McGraw Hill, New Delhi, 2008.

6. Mittal Ayush: Laplace Transform and Matrix Decomposition Based Cryptographic Scheme, Journal of Xi'an University of Architecture & Technology, ISSN No : 1006-7930, Volume XII, Issue II, 2020, pp. 551 - 560.
7. Samriddhi V, Sanjana, Shalini, Sinchana A, Suma V Shetty: XOR Cipher, International Journal on Science and Technology (IJSAT), E-ISSN: 2229-7677, Volume 16, Issue 1, January-March 2025.
8. Stallings W.: Cryptography and Network Security: Principles and Practices, Prentice Hall, 1999.
9. Yegnanarayanan V., Yakkala Poojitha: Pythagorean Triples in Cryptography and Associated Networks, International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075 (Online), Volume-8 Issue-12, October 2019.