JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

"A Novel Cryptographic Approach Using Matrices, ASCII Encoding, and Octal Representation"

[1]Mr. Ranjeet Patil, [2]Amitkumar Hanamantrao Jagadale, [3]Ganesh Babar

[1]Yashoda Technical Campus,Satara, [2] Yashoda Technical Campus,Satara, [3]Yashoda Technical Campus,Satara
[1]Department of Basic Science and Humanity,
[1]Yashoda Technical Campus,Satara,Satara, Maharashtra, India.

Abstract: This research introduces a new method of cryptography that uses matrices, ASCII values, and the octal number system to make data more secure. Matrices help to transform data in a way that is difficult for hackers to reverse. ASCII values allow text to be converted into numbers, making it easier to use in mathematical operations. The octal number system provides a compact way to represent data, saving space and adding an extra layer of security. By combining these techniques, we have created encryption and decryption methods that are both strong and efficient. This approach ensures that data stays safe from common hacking methods while being practical to use in real-world applications.

KEYWORDS: cryptography, ASCII values, octal number system, Matrices, encryption and decryption

I. INTRODUCTION

The widespread adoption of the internet, e-commerce, social networking, and cloud-based platforms has revolutionized the way information is shared, stored, and processed. While these advancements offer significant benefits, they also introduce critical security challenges, including unauthorized access, fraud, identity theft, and large-scale cyber-attacks. In this digital environment, protecting sensitive data during transmission and storage has become a vital necessity for individuals, businesses, and government institutions. Network security and cryptography together form the foundation for ensuring the confidentiality, integrity, and availability of information in both public and private communication systems. Network security applies a combination of software, hardware, policies, and configurations to safeguard data, while cryptography secures messages by transforming readable plaintext into unintelligible cipher text that can only be recovered using a valid decryption key. In addition to encryption and decryption, related concepts such as hashing, steganography, and cryptanalysis contribute to the broader landscape of information protection—ensuring data authenticity, preventing tampering, and evaluating the strength of cryptographic systems. There are many encryption algorithms that have been developed and used over time. Symmetric key techniques, such as DES and AES, rely on a single key for both encryption and decryption, offering high-speed performance. Asymmetric key algorithms like RSA and ECC use distinct public and private keys, enabling secure key exchange and digital signatures. Hash functions such as MD5 are widely used for data integrity verification. Alongside these established methods, researchers continue to explore innovative approaches that enhance security and computational efficiency. Examples include hexadecimal swapping with reversal operations on plaintext and keys, SNS with symbol remapping, and the Lucas Polynomial Cryptosystem, which leverages the unique properties of Lucas numbers for encryption and decryption. These novel techniques add new layers of complexity to cipher text generation, improving resistance to cryptanalysis and reducing vulnerability to attacks. Despite continuous research, implementing cryptographic innovations in real-world systems remains challenging. Factors such as miscommunication between stakeholders, usability constraints, unclear responsibility boundaries, and conflicting priorities often hinder adoption. Addressing these barriers requires strong collaboration between cryptographers, software developers, and standardization bodies to ensure that secure systems are practical, user-friendly, and widely deployable. This paper provides a comprehensive study of network security and cryptography, covering both conventional and emerging encryption–decryption algorithms. It compares their performance, particularly in terms of the time required to encrypt and decrypt files of varying sizes,

and examines their respective security strengths and weaknesses. By combining foundational principles with advanced cryptographic techniques, this work aims to contribute toward the development of robust, efficient, and adaptable security solutions for the modern digital landscape.

Literature Review :

Dr. Ummadi Thirupalu and colleagues [1] presented a hexadecimal swapping approach in which the hexadecimal sequences of both plaintext and encryption keys are reorganized. Such rearrangement improves the cryptographic properties of confusion and diffusion, reflecting the principles outlined by Shannon. In a related vein, research on Hexadecimal Character Substitution Ciphers (HCSC) and swapping-oriented encryption in image protection has shown that even relatively simple data-level changes can yield secure encryption while keeping processing demands low. Taken together, these works signal a noticeable shift in cryptographic design towards combining inventive transformation methods with conventional algorithms, aiming to produce solutions that remain both efficient and resilient.

Debasis Das and co-authors [2] presented what they term the Strange Number System (SNS), a scheme built on octovicentimal base conversion combined with symbol remapping and flexible algorithmic routines. The approach is designed to perform both encryption and decryption in a way that is secure yet computationally practical. In essence, SNS operates as an encoding converter for textual data, delivering a degree of protection that rivals, and in certain aspects may exceed, the security of a one-time pad. Notably, it mitigates the limitations found in fixed encryption methods, which are often vulnerable to brute-force attacks, thereby positioning itself as a promising and resilient option for contemporary cryptographic use.

Gudela Ashok and colleagues [3] worked on what they refer to as the Lucas Polynomial Cryptosystem. The method takes ideas from Lucas numbers and mixes them with polynomial structures, with the aim of finding a balance between solid mathematical design and reasonable processing needs. It's still in the early testing phase, so there's a lot that isn't known yet. Even so, the first results look promising, and if the approach holds up in further studies, it might add a useful option for keeping data safe and private in modern digital setups.

Pronika and S. S. Tyagi [4] ran a set of tests on different encryption algorithms—DES, Blowfish, ARC4, DES3, and AES—to see how each performs under different conditions. What they found was that speed and efficiency aren't fixed traits; they change depending on how the key is structured, the type of cryptography, and even the file size. In their results, DES handled very small files, about 1 KB, better than the rest, while ARC4 was faster for much larger files, close to 1 MB. AES wasn't the quickest because it demands more computing power, but it stood out for allowing variable key lengths and for being widely trusted for its security.

In another study, Ekta Agrawal and her team [5] suggested a lightweight way to encrypt short messages. Their method works by using the length of the message in the encryption process, which cuts down on the time it takes. Right now, it's focused on plain text, though they plan to adapt it for special characters so it can be used for more types of messages.

Kinjal Raut and co-authors [6] studied a set of widely used cryptographic algorithms and compared how they perform. From their results, it was clear that symmetric approaches tend to run faster than asymmetric ones in most situations. Among these, AES proved to be the most dependable option, managing to combine strong encryption with flexible key sizes and reasonable processing times.

Konstantin Fischer and his team [7] tried to figure out why so much cryptography research doesn't end up as something people can actually use in secure products. They came across a number of reasons. Sometimes researchers and developers are simply aiming at different things, so their work doesn't line up. Getting agreement on standards can also drag on for quite a while. On top of that, some of the reference versions available just aren't strong enough to be used in practice. They suggest that being more open, working more closely across fields, and giving clear, hands-on advice could help close the gap between research and real-world use.

Ms. S. Anitha and her team [8] wrote about how crucial cryptography is when it comes to securing networks and cloud systems. They mentioned that in recent years, there's been a shift toward using more advanced mathematics and multiple key setups to keep information private and make sure the data stays accurate. But even with those improvements, handling the keys and exchanging them safely is still a big challenge. The authors believe that future work should put more effort into developing encryption methods that fit the needs of cloud computing, where speed, reliability, and strong protection all have to be balanced.

M. Divya [9] talked about how important cryptography is for keeping online communication secure, especially now that older methods are much easier to break. She explained that if only the sender and the receiver hold the encryption keys, it becomes far harder for anyone else to get into the data. This also works as a way to check who the user really is, adding extra protection.

R. Janani [10] observed that protecting data exchanged over the internet depends heavily on effective use of cryptography. She argued that keeping encryption keys completely confidential is essential, as those keys not only make secure exchanges possible but also enable the verification of client identities. In her assessment, both security and authentication have become basic expectations in today's networked world rather than optional safeguards.

Prerna Sharma and colleagues [11] explored different strategies for improving security in network communication. They didn't just treat cryptography as a standalone subject; instead, they looked at how it plays a role while data is actually being transmitted. Their study noted that combining cryptography with steganography can make systems more resilient. This mix not only protects the information but can also make it less obvious to an attacker that any valuable data is being sent in the first place.

RESEARCH METHODOLOGY:

I] Covert an ASCII value to an octal number contains two-way

Convert the character to its ASCII value (decimal value). Each character has a corresponding ASCII value (decimal value) in the table. These are as follows

Sr No.	Capital Character	ASCII Value	Small Character	ASCII Value
1	A	65	a	97
2	B	66	b	98
3	C	67	c	99
4	D	68	d	100
5	E	69	e	101
6	F	70	f	102
7	G	71	g	103
8	H	72	h	104
9	I	73	i	105
10	J	74	j	106
11	K	75	k	107
12	L	76	l	108
13	M	77	m	109
14	N	78	n	110
15	O	79	o	111
16	P	80	p	112
17	Q	81	q	113
18	R	82	r	114
19	S	83	s	115
20	T	84	t	116
21	U	85	u	117
22	V	86	v	118
23	W	87	w	119

24	X	88	x	120
25	Y	89	y	121
26	Z	90	z	122

II] Convert the decimal value to octal number.

The division-by-8 method is the most popular way to convert a value in decimals to its octal equivalent.

.

- This method involves repeatedly dividing the decimal number by 8 and recording the remainder at each stage.
- This remainder will be one of the octal integers.
- Take the quotient from the former division and divide it by 8 again.
- Record the new remainder.
- duplication this process until the quotient becomes zero.
- The octal number is formed by writing the remainders by hamper (reverse) order.

Example

ASCII value of A is 65.

Decimal to octal

65 ÷ 8 quotient 8 & remainder 1
8 ÷ 8 quotient 1 & remainder 0
1 ÷ 8 quotient 0 & remainder 1
Octal number is 101.

III] Covert the Octal Number to Decimal Number

Using the following formula

$$\sum_{i=0}^{i=n-1} d_i 8^i$$

Where $d_0, d_1, d_2 \ldots\ldots\ldots\ldots d_n$ *are the digits of the octal number from right to left.*

Example

Octal number 110

Using above formula we get

$$8^0 \times 0 + 8^1 \times 1 + 8^2 \times 1 = 0 + 8 + 64 = 72$$

72=H

IV] FINDING THE INVERSE OF A 4×4 MATRIX USING THE CAYLEY-HAMILTON THEOREM

Determining the Characteristic Polynomial is the first step. Design the matrix $A-\lambda I$ A−λI, where I I is the 4×4 identity matrix and λ λ is a scalar.

The factor |A−λI|=0 was calculated.

Consequently, a fourth-degree polynomial in λ, the characteristic polynomial, is generated:

$$\boldsymbol{\lambda^4 + a_1\lambda^3 + a_2\lambda^2 + a_3\lambda + a_4 = 0}$$

The Cayley-Hamilton Theorem's application in step two

The matrix A fulfills a specific characteristic equation for the Cayley-Hamilton theorem. Consequently,

$$A^4 + a_1A^3 + a_2A^2 + a_3A + a_4\mathrm{I} = 0 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\mathbf{1}$$

The inverse of A must be expressed using this equation.

Step Three: The result of multiplying equation 1 by $\boldsymbol{A^{-1}}$ we get

$$\boldsymbol{A^3 + a_1A^2 + a_2A + a_3I + a_4A^{-1} = 0}$$

So,

$$\boldsymbol{A^{-1} = -\frac{1}{a_4}(A^3 + a_1A^2 + a_2A + a_3I)}$$

Determine powers $\boldsymbol{A^3}, \boldsymbol{A^2}$ *etc.*

$$\boldsymbol{A^2 = A * A}$$
$$\boldsymbol{A^3 = A^2 * A}$$

Replace all terms that is A^3, A^2& I within the following equation

$$A^{-1} = -\frac{1}{a_4}(A^3 + a_1A^2 + a_2A + a_3\mathrm{I})$$

We obtain the necessary inverse

Note: This techniques only is effective when A has non-singular i.e., det(A)≠0, or similarly, a4≠0 in the characteristic polynomial.

Matrix Encoding by using above methods

Example : Encode this message

WELCOME TO BHADOLEHI.

ASCII value Matrix is A=$\begin{bmatrix} W & O & T & H & L \\ E & M & O & A & E \\ L & E & Space & D & H \\ C & Space & B & O & I \end{bmatrix} = \begin{bmatrix} 87 & 79 & 84 & 72 & 76 \\ 69 & 77 & 79 & 65 & 69 \\ 76 & 69 & 0 & 68 & 72 \\ 67 & 0 & 66 & 79 & 73 \end{bmatrix}$

Convent Matrix A to Octal Number

O=$\begin{bmatrix} 127 & 117 & 124 & 110 & 114 \\ 105 & 115 & 117 & 101 & 105 \\ 114 & 105 & 0 & 104 & 110 \\ 103 & 0 & 102 & 117 & 111 \end{bmatrix}$

B is any invertible Matrix

B=$\begin{bmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Inverse of B =B^{-1}

$$B^{-1} = \begin{bmatrix} 1 & -2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

M=B*O which is meaningless Matrix

M=$\begin{bmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 127 & 117 & 124 & 110 & 114 \\ 105 & 115 & 117 & 101 & 105 \\ 114 & 105 & 0 & 104 & 110 \\ 103 & 0 & 102 & 117 & 111 \end{bmatrix}$

M=$\begin{bmatrix} 337 & 347 & 358 & 312 & 324 \\ 105 & 115 & 117 & 101 & 105 \\ 114 & 105 & 0 & 104 & 110 \\ 103 & 0 & 102 & 117 & 111 \end{bmatrix}$

Which is meaningless Matrix.

Matrix Decode

To convert this meaningless matrix to Octal Number matrix Multiply this matrix by B^{-1}

We get.

O=$B^{-1}M$

O=$\begin{bmatrix} 1 & -2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 127 & 117 & 124 & 110 & 114 \\ 105 & 115 & 117 & 101 & 105 \\ 114 & 105 & 0 & 104 & 110 \\ 103 & 0 & 102 & 117 & 111 \end{bmatrix}$

O=$\begin{bmatrix} 127 & 117 & 124 & 110 & 114 \\ 105 & 115 & 117 & 101 & 105 \\ 114 & 105 & 0 & 104 & 110 \\ 103 & 0 & 102 & 117 & 111 \end{bmatrix}$

To convert Octal to ASCII Value using above method we get

$$\begin{bmatrix} 87 & 79 & 84 & 72 & 76 \\ 69 & 77 & 79 & 65 & 69 \\ 76 & 69 & 0 & 68 & 72 \\ 67 & 0 & 66 & 79 & 73 \end{bmatrix} = \begin{bmatrix} W & O & T & H & L \\ E & M & O & A & E \\ L & E & Space & D & H \\ C & Space & B & O & I \end{bmatrix} = A$$

WELCOME TO BHADOLEHI.

Implementation

```
import numpy as np

# Step 1: Original message
message = "WELCOME TO BHADOLEHI."

# Step 2: Create ASCII value matrix (4x5 matrix as in example)
ascii_matrix = np.array([
    [87, 79, 84, 72, 76],
    [69, 77, 79, 65, 69],
    [76, 69, 0, 68, 72],
    [67, 0, 66, 79, 73]
])

print("ASCII Matrix:\n", ascii_matrix)

# Step 3: Convert ASCII to Octal (string representation for clarity)
octal_matrix = np.vectorize(lambda x: int(oct(x)[2:]))(ascii_matrix)
print("\nOctal Matrix:\n", octal_matrix)
```

```
ASCII Matrix:
 [[87 79 84 72 76]
 [69 77 79 65 69]
 [76 69  0 68 72]
 [67  0 66 79 73]]

Octal Matrix:
 [[127 117 124 110 114]
 [105 115 117 101 105]
 [114 105   0 104 110]
 [103   0 102 117 111]]

Encrypted Matrix (M):
 [[337. 347. 358. 312. 324.]
 [105. 115. 117. 101. 105.]
 [114. 105.   0. 104. 110.]
 [103.   0. 102. 117. 111.]]
```

```
# Step 4: Define encryption matrix B and its inverse
B = np.array([
    [1, 2, 0, 0],
    [0, 1, 0, 0],
    [0, 0, 1, 0],
    [0, 0, 0, 1]
], dtype=float)

B_inv = np.linalg.inv(B)

# Step 5: Encrypt → M = B * O
M = np.dot(B, octal_matrix)
print("\nEncrypted Matrix (M):\n", M)

# Step 6: Decrypt → O = B_inv * M
O_recovered = np.dot(B_inv, M)
print("\nRecovered Octal Matrix:\n", O_recovered)

# Step 7: Convert Octal back to ASCII values
ascii_recovered = np.vectorize(lambda x: int(str(int(round(x))), 8
    ))(O_recovered)

print("\nRecovered ASCII Matrix:\n", ascii_recovered)

# Step 8: Convert ASCII values back to characters
decoded_message = "".join(chr(val) if val != 0 else " " for val
    ascii_recovered.flatten())
print("\nDecoded Message:\n", decoded_message)
```

```
Recovered Octal Matrix:
 [[127. 117. 124. 110. 114.]
 [105. 115. 117. 101. 105.]
 [114. 105.   0. 104. 110.]
 [103.   0. 102. 117. 111.]]

Recovered ASCII Matrix:
 [[87 79 84 72 76]
 [69 77 79 65 69]
 [76 69  0 68 72]
 [67  0 66 79 73]]

Decoded Message:
 WOTHLEMOAELE DHC BOI

=== Code Execution Successful ===
```

Conclusion

This research presented a cryptographic technique that integrates matrices, ASCII values, and the octal number system to enhance data security. By combining these mathematical tools, the proposed method introduces additional layers of complexity, making unauthorized decryption more difficult while maintaining computational efficiency. The approach not only secures data through systematic conversion processes but also demonstrates how traditional number systems and matrix operations can be effectively applied in modern cryptography. Experimental examples illustrate the feasibility of the method, showing that meaningful messages can be successfully encrypted into unreadable forms and later recovered with accuracy. However, certain limitations need to be acknowledged. The reliance on ASCII values restricts the current model mainly to text data, limiting its adaptability for multimedia files such as images or audio. The use of matrix inversion also requires that the chosen matrix be non-singular, which imposes constraints on key selection. Additionally, while the octal system adds compactness and an extra layer of security, it may not offer the same robustness as more complex number systems when exposed to advanced cryptanalytic attacks. Practical implementation challenges, such as processing overhead for very large files and integration with existing cryptographic standards, also remain areas for further research. Future work can focus on extending this method to support diverse data types, testing its resistance against modern attack models, and optimizing the algorithm for real-time applications. With these improvements, the proposed system has the potential to evolve into a reliable and adaptable encryption scheme suitable for real-world deployment.

Limitations

1. The method is currently limited to text data, as it relies on ASCII values and does not directly support multimedia files such as images, audio, or video.
2. Successful encryption and decryption depend on the use of a non-singular matrix; if the determinant is zero, the process cannot be applied.
3. The use of the octal number system, while compact, may not provide the same level of cryptographic strength as more advanced or hybrid number systems.
4. For very large datasets or high-speed applications, the matrix operations may introduce computational overhead, affecting efficiency.
5. Integration with existing cryptographic frameworks and standards remains a challenge, which could limit its direct adoption in real-world systems.
6. The security of the method has not yet been extensively tested against advanced attack models such as differential or linear cryptanalysis.

REFERENCES

[1] Thirupalu, U., Padmavathi Devi, S. V., & Chandrakanth, P. (2024, August). A simple technique to generate ciphertext using hexadecimal swapping on both the plaintext and the key. *International Research Journal of Modernization in Engineering Technology and Science, 6*(8).

[2] Das, D., & Lanjewar, U. A. (2012, March–April). Strange number system: An enhancing tool for data encryption and decryption. *International Journal of Advanced Research in Computer Science, 3*(2).

[3] Ashok, G., Kumar, S. A., & Kumari, D. C. (2023, August). Lucas polynomial cryptosystem: A novel approach for secure encryption. *Journal of Harbin Engineering University, 44*(8).

[4] Pronika, & Tyagi, S. S. (2021, August). Performance analysis of encryption and decryption algorithm. *Indonesian Journal of Electrical Engineering and Computer Science, 23*(2), 1030–1038.

[5] Agrawal, E., & Pal, P. R. (2017, May). A secure and fast approach for encryption and decryption of message communication. *International Journal of Engineering Science and Computing, 7*(5).

[6] Raut, K., & Katkar, C. (2021, December). A comprehensive review of cryptographic algorithms. *International Journal for Research in Applied Science & Engineering Technology, 9*(12).

[7] Fischer, K., Trummová, I., Gajland, P., Acar, Y., Fahl, S., & Sasse, M. A. (2024). The challenges of bringing cryptography from research papers to products: Results from an interview study with experts (extended version). In *Proceedings of the USENIX Security Symposium 2024*.

[8] Anitha, S., & Padmalatha, R. (2022, January). A study on network security and cryptography. In *AICTE Conference Proceedings*.

[9] Divya, M. (2023, January). Role of cryptography in securing online communication. *International Research Journal of Modernization in Engineering Technology and Science, 5*(1).

[10] Janani, R. (2022, February 8). Importance of cryptography in data security over the internet. *EasyChair Preprint*.

[11] Sharma, P., Yadav, K., & Tiwari, A. K. (2022, May). A review paper on network security and cryptography. *World Journal of Research and Review, 14*(5), 20–24.