Tetrad's Cryptography Algorithm For Enhancing Data Security Along Side Lost Data Retrieval

Dr. P. Sriram Chandra[1], V. Bala Shankar[2], N. Venkatesh[3*], P. Pushkal RamTej[4], B.S.N.Ganesh[5] and T. Puruhuthika[6]

[1] Professor & HOD, [2]Assistant Professor, [1]Department of CSE-DS
[2 3 4 5 6] Department of Computer Science and Engineering (AIML & CS)
[123456]Godavari Institute of Engineering & Technology, Rajahmundry, Andhra Pradesh, India
[1]dr.psr@yahoo.com, [2]balasankar@giet.ac.in,
[3*]venkateshnalla21@gmail.com, [4]pushkalramtej123@gmail.com,
[5]bsnganesh2002@gmail.com, [6]tangellapuruhuthika@gmail.com

Abstract. TETRAD'S represents an exceptional cryptographic algorithm innovated by the authors, with the objective of enhancing the security of data. This algorithm employs matrix representations, XOR operations on binary strings, and customized functions to generate an S-box for encryption. In addition, the algorithm generates a key to heighten security and provides the capability to recover messages and decrypt encrypted data. Along with security, this paper contributes to retrieval of lost data. Techniques for data recovery, such as the utilization of specialized forensic instruments and tools such as a hex editor and Foremost Tool, are employed to restore data that has been lost or corrupted. The proposed security algorithm and data retrieval system in this article exemplifies efficient resource utilization, consistent performance, expedient data retrieval, elevated encryption and decryption throughput, and rapid encryption and decryption durations. The study also underscores the significance of equipping digital forensic analysts with tools that facilitate data recovery.

Keywords: Data Security, Data and File Recovery System, Encryption and Decryption, Foremost and Hex editor.

1 Introduction

The paper focuses on addressing critical concerns in cybersecurity and computer forensics, emphasizing data security and recovery in the digital landscape. It introduces a novel system utilizing the TETRAD cryptographic method, renowned for robust encryption, as the cornerstone for safeguarding data. Enhancements to the TETRAD method include the incorporation of features like S-box formation, binary conversion, and XOR operations, enhancing the encryption process and key generation mechanism. The paper not only excels in encrypting data effectively but also provides a reliable data recovery solution. For data recovery, the paper employs the proven effectiveness of a hex editor and Foremost Tool, renowned forensic tools. These tools operate within the LINUX Command Line Interface, ensuring thorough and trustworthy data recovery.

K. R. Madhavi et al. (eds.), *Proceedings of the International Conference on Computational Innovations and Emerging Trends (ICCIET 2024)*, Advances in Computer Science Research 112,
https://doi.org/10.2991/978-94-6463-471-6_136

The overarching goal is to present a holistic approach to protect and restore digital information amidst the complexities of cybersecurity and computer forensics.

2 Background study

Akanksha Mathur introduced a novel encryption and decryption method that hinges on the ASCII values of characters within the plaintext [1]. Classified under symmetric key encryption, this algorithm employs the same key for both encryption and decryption, functioning optimally when the input data's length aligns with that of the key [1].

The RSA algorithm, pioneered by Ron Rivest, Adi Shamir, and Leonard Adleman, stands out as a widely adopted and well-established technique in asymmetric key cryptography [2]. In this cryptographic paradigm, distinct public and private keys play pivotal roles in encryption and decryption processes, respectively [3]. Digital signatures, a method where the sender electronically signs data, signify ownership of a private key corresponding to a publicly announced public key [4]. The associated security principle, nonrepudiation, prevents either the sender or receiver from denying involvement in a data transaction [5, 6]. Anjula Gupta's research delves into the historical roots and significance of cryptography, introducing a spectrum of asymmetric algorithms contributing to security and privacy realms [7].In the contemporary landscape, data security has become paramount across all sectors [8]. Various deployed techniques and algorithms aim to safeguard data accessibility, privacy, and integrity from threats and breaches [8]. Security, as a concept, spans dimensions such as confidentiality, integrity, and availability [9]. Data recovery entails retrieving data from both allocated and unallocated spaces [10]. Allocated space contains currently accessible files that can be logically read, while unallocated space includes files no longer accessible, even if deleted from storage, and cannot be read in a logical manner [11,12].Data loss, the corruption, or inadvertent deletion of previously owned data, underscores the critical need to equip digital forensic analysts with tools for recovery [13, 14].The study's focus on the architecture of the NTFS file system stems from its widespread global utilization, meeting the needs and preferences of countless users [15].

3 Proposed Methodology

The TETRAD’S Cryptographic Algorithm enhances data security through a sequence of stages, encompassing key generation, encryption, and decryption. It integrates unique components like the creation of S-boxes, binary transformations, matrix operations, and XOR manipulations, offering robust protection.

3.1 TETRAD’S Cryptographic Algorithm

In this section, the three fundamental components of the newly proposed cryptographic algorithm: Key Generation, Encryption, and Decryption were briefly explored.

Key Generation. The process involves several significant steps:

Padding. To ensure the binary data is a multiple of a specific number (N), additional bits are introduced. The padding calculation, expressed as,

$$P = L + (N - (L \bmod N)) \qquad (1)$$

Here, (P) is the padded length, (L) is the original length, and (N = 4) is the designated number.

Dividing. The padded binary data is partitioned into uniform chunks of (N) bits each, treating them as base 2 integers. The dividing formula defines the number of chunks.

$$(D = P/N) \qquad (2)$$

Here, D is the chunk count, P is the padded length, and N is the predefined number.

Matrix Conversion. Each binary chunk transforms into an NxN matrix, with elements as 0 or 1.

$$Mn = Pn\, x\, N \qquad (3)$$

where P_n represents individual chunks, n denotes the sequence, and N is the predefined number.

Row and Column Interchange. Rows and columns within each matrix undergo swapping based on a predetermined pattern, determined by a permutation function or encryption algorithm. The interchange operation is denoted as R ↔ C.
where R represents rows, and C represents columns in each matrix.

Swapping and Conversion. Positions of adjacent matrices are swapped, followed by converting matrices back to binary format before combining them.

Circular Right Shift. Each matrix element experiences a circular right shift by one position, wrapping the last element to the first position.

Loop. The above operations, commencing from the dividing step, are iterated L/4 times.

S-Box. Bit substitution occurs using a pre-defined S-Box after completing the loop process.

Division and XOR Operation. The bits are divided into two parts, and an XOR operation is applied to yield the final key.

The final key input is obtained by concatenating all the shifted matrices in order.

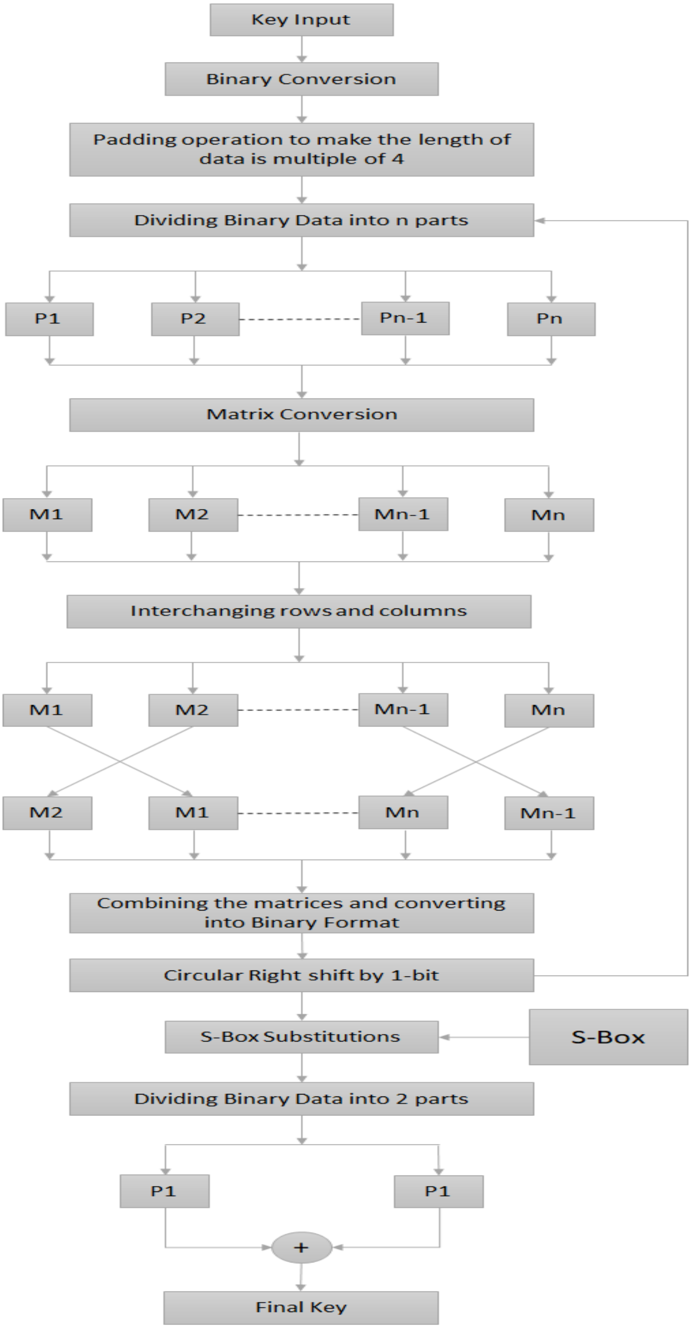

Key Input
Binary Conversion
Padding operation to make the length of data is multiple of 4
Dividing Binary Data into n parts
P1
P2
Pn-1
Pn
Matrix Conversion
M1
M2
Mn-1
Mn
Interchanging rows and columns
M1
M2
Mn-1
Mn
M2
M1
Mn
Mn-1
Combining the matrices and converting into Binary Format
Circular Right shift by 1-bit
S-Box Substitutions
S-Box
Dividing Binary Data into 2 parts
P1
P1
+
Final Key

Fig. 1. Block Diagram of Proposed Algorithm Key Generation

The figure 1 shows a key generation process that involves several steps of binary conversion, matrix manipulation, circular shift, S-box substitution, and XOR operation. The process aims to enhance the security of the key by introducing non-linearity and diffusion.

Encryption. The Tetrad Cryptographic Algorithm is a fast and secure algorithm that encrypts binary data using a 9-round encryption algorithm:

The Tetrad Cryptographic Algorithm is a symmetric block cipher that encrypts and decrypts data using a binary key and a 9-round encryption algorithm. The algorithm involves several steps of binary conversion, XOR operation, block division, initialization vector, substitution, shift rows, mix columns, and concatenation.
The key in the Tetrad Cryptographic Algorithm is represented as a binary string of length N, where N is a predefined number. The key is formulated as

$$K = S1 \,||\, S2 \,||\ldots||\, Sn \tag{4}$$

where K denotes the key, *Si* represents the i-th shifted matrix, and || signifies concatenation.
For binary conversion and XOR Key, the data undergoes conversion into binary format through an encoding scheme like ASCII, UTF-8, Base64, etc. The key is then XORed with the binary data, expressed as

$$B = E(D) \oplus K \tag{5}$$

Here, B stands for binary data, D denotes the original data, K is the Key, ⊕ indicates bitwise XOR, and E represents the encoding function.
Data is further divided into blocks, with each block being (N) bits in size. These blocks are treated as integers in base 2, following the formula

$$(Bi \;=\; B/N) \tag{6}$$

where (Bi) represents the i-th block, (B) is the binary data, and (N) is the predefined number.
Incorporating a random initialization vector (IV) of N bits, generated using a random function *R*(*N*), the first block undergoes a bitwise XOR operation with the IV. This substitution operation replaces each bit of the block with the opposite bit of the IV, represented as

$$IV = R(N) \tag{7}$$

&

$$B1' = B1 \oplus IV \tag{8}$$

where IV is the initialization vector, R is the random function, $1B1$ is the first block, $1'B1'$ is the substituted block, and ⊕ signifies bitwise XOR.

The encryption process involves 9 rounds, each encrypting a block using a 9-round encryption algorithm. Each round comprises three operations: substitution, shift rows, and mix columns. These operations share similarities with the AES round functions but employ different constants and operate on N bits instead of 128 bits. The formula for the 9-round encryption algorithm is

$$Ci = E(Bi) \oplus Bi \qquad (9)$$

where Ci represents the i-th ciphertext block, Bi is the i-th plaintext block, and E is the encryption function applying nine rounds of substitution, shift rows, and mix columns to the block.

The Tetrad Cryptographic Algorithm's output is the concatenation of the ciphertext blocks, denoted as

$$\text{Output} = (C1 \,||\, C2 \,||\, \ldots \,||\, C\,n) \qquad (10)$$

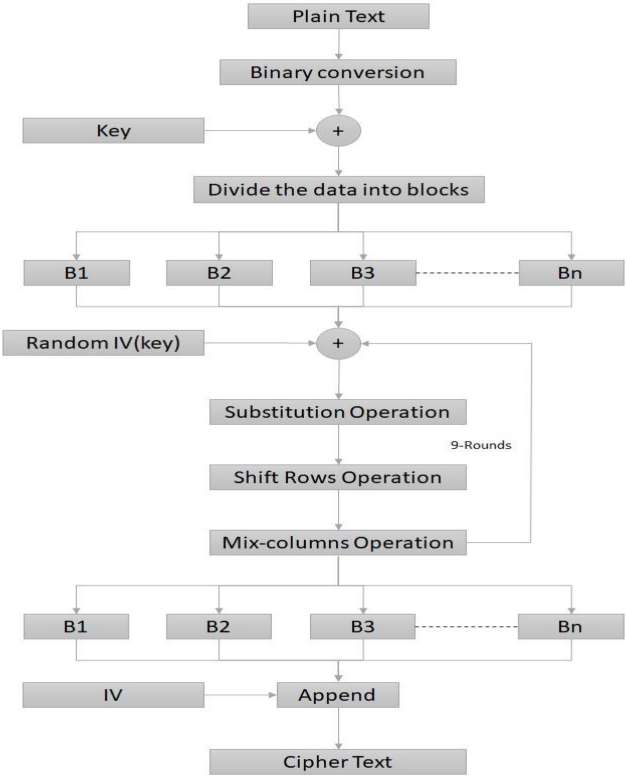


Fig. 2. Block Diagram of Proposed Encryption Algorithm

The Fig 2 shows Tetrad Cryptographic Algorithm encrypts data by converting plaintext to binary, dividing it into blocks, and applying a 9-round process of substitution, shifting, and mixing. The output is a secure ciphertext generated from concatenated encrypted blocks.

Decryption. In Tetrad's Cryptographic Algorithm, the Decryption process is similar to the Encryption that is reverse process of Encryption. The process is as follows:

The ciphertext is converted into binary format using some decoding scheme. Then the initialization vector (IV) is detached from the first block of binary data using bitwise XOR operation. Later the binary data is divided into blocks of N bits each, where N is a predefined number. Then each block is decrypted using a 9-round decryption algorithm that consists of three operations: substitution, shift rows, and mix columns. Finally, the Key is appended to get the plaintext from the ciphertext.

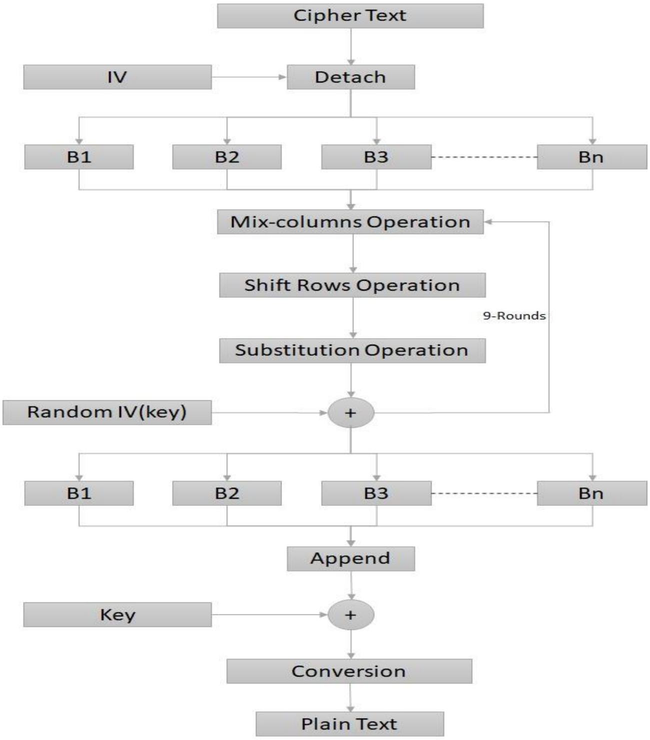
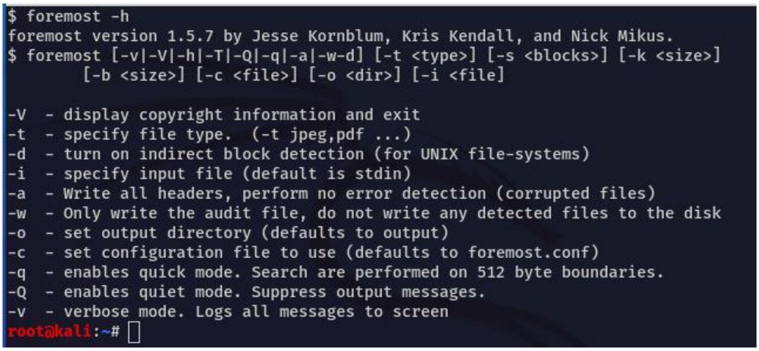


Fig. 3. Block Diagram of Proposed Decryption Algorithm

The fig 3 shows depict the decryption process of cipher text into plain text, involving detachment into blocks, 9 rounds of mix-columns, shift rows, and substitution operations, followed by appending a random IV and key conversion.

3.2 Data Recovery

Foremost, a command-line tool recognized for its file carving capabilities, is frequently employed to extract files from block devices or damaged storage media without relying on filesystem metadata. Particularly valuable for recovering files from disk images, Foremost offers an effective solution in situations where conventional file recovery methods may prove inadequate.



```
$ foremost -h
foremost version 1.5.7 by Jesse Kornblum, Kris Kendall, and Nick Mikus.
$ foremost [-v|-V|-h|-T|-Q|-q|-a|-w-d] [-t <type>] [-s <blocks>] [-k <size>]
        [-b <size>] [-c <file>] [-o <dir>] [-i <file]

-V  - display copyright information and exit
-t  - specify file type.  (-t jpeg,pdf ...)
-d  - turn on indirect block detection (for UNIX file-systems)
-i  - specify input file (default is stdin)
-a  - Write all headers, perform no error detection (corrupted files)
-w  - Only write the audit file, do not write any detected files to the disk
-o  - set output directory (defaults to output)
-c  - set configuration file to use (defaults to foremost.conf)
-q  - enables quick mode. Search are performed on 512 byte boundaries.
-Q  - enables quiet mode. Suppress output messages.
-v  - verbose mode. Logs all messages to screen
root@kali:~# 
```

Fig. 4. Foremost Tool Interface

To initiate Foremost, users can utilize the following command:

```
foremost -i /path/to/image.dd -t jpg,png,pdf
```

In this command, the '-i' flag specifies the input file or disk image for file recovery, with '/path/to/image.dd' indicating the path to the disk image being analyzed. The '-t' flag is utilized to designate the types of files targeted for recovery, with this instance configured to retrieve JPEG, PNG, and PDF files.

Verbose Mode, indicated by the '-v' flag, provides more intricate details during the file recovery process. Enabling Verbose Mode results in supplementary information about the progress of recovery, encompassing the types of files undergoing processing, their sizes, and any encountered issues.

Foremost also supports the utilization of configuration files, empowering users to customize various settings for the file recovery process. This encompasses specifying file types, adjusting block sizes, and configuring output directories.

The subsequent command illustrates the utilization of a configuration file:

```
foremost -i /path/to/image.dd -c foremost.conf
```

Utilizing a configuration file augments flexibility, allowing users to tailor Foremost to meet their specific recovery requirements.

4 Testing and Results

During the testing and results phase, the TETRAD'S Cryptographic Algorithm undergoes thorough examination. Key parameters, encompassing key generation, encryption, and decryption times, shed light on its efficiency. Simultaneously, the data recovery methodology, employing Foremost, is evaluated for file carving efficiency and recovery time. The comprehensive analysis of TETRAD'S includes crucial factors like time consumption, throughput, and memory usage, offering a succinct overview of its performance and resilience in securing data and facilitating efficient recovery.

4.1 TETRAD'S Algorithm

In the comprehensive testing and subsequent analysis conducted for the proposed Tetrad Cryptographic Algorithm, the results depicted in table 1 revealed a notable superiority over established algorithms such as AES and DES across key parameters. The encryption and decryption speeds of Tetrad were particularly remarkable, demonstrating milliseconds far lower than those of AES and DES. The throughput analysis further underscored Tetrad's efficiency, consistently outperforming both algorithms in terms of data transfer rates. One of the most striking findings pertained to memory usage, where Tetrad showcased a significantly smaller footprint, ranging from 145 to 177 bytes, in contrast to the kilobytes required by AES and DES. This observation positions Tetrad as not only a faster and more throughput-efficient alternative but also as a memory-efficient solution for cryptographic operations. These collective results substantiate Tetrad's potential as a superior choice for a range of cryptographic applications, balancing the critical aspects of speed, efficiency, and security. In practical terms, Tetrad emerges as a promising solution for scenarios demanding rapid and resource-efficient cryptographic processes without compromising on robust security measures.

Table 1. A Comparison of Encryption Algorithms: AES, DES, and TETRAD

Parameters	AES	DES	TETRAD
Encryption (MS)	0.715575	0.815545	0.001002
Decryption (MS)	0.00015	0.00018	0.000132
Encryption Throughput	0.74 Mbps to 1.19 Mbps.	0.73 Mbps to 1.18Mbps.	0.76Mbps to 1.20 Mbps.
Decryption Throughput	5.809 Mbps to 23.273	5.205 Mbps to 23.273	5.809 Mbps to 23.273
Memory Usage	1 to 10 kilobytes	1 to 5 kilobytes	145 to 177 bytes

4.2 Data Recovery

The table 2 data present a comprehensive overview of the recuperation status for JPG, PNG, and MP4 files in unallocated spaces employing four distinct recovery instruments. Tsk Recover demonstrates a reasonable degree of success, attaining a 50% recovery rate for JPG and PNG files and a superior success rate of 75% for MP4 files. Ftk Imager exhibits marginally improved outcomes, with a 60% recovery rate for PNG files and a 65% success rate for MP4 files. Foremost emerges as the most potent tool, boasting an impeccable recovery rate for all file formats. Similarly, Hex Editor proves highly triumphant, accomplishing an unblemished 100% recovery rate across JPG, PNG, and MP4 files. These findings underscore the varying effectiveness of recovery tools, with Foremost and Hex Editor exemplifying outstanding performance in the retrieval of data from unallocated spaces.

Table 2. A Comparison of Recovery Tools

No	Tools	Status of Recovery		
		JPG	PNG	MP4
		Unallocated Space		
1	Tsk Recover	50%	50%	75%
2	Ftk Imager	50%	60%	65%
3	Foremost & Hex editor	65%	70%	80%

While the foremost tool, hex editor can recover better than other tools . However, tools that can't recover completely don't mean they're not good. These tools are still recommended and can be used to assist investigators in the investigation process. Investigators can have several options for forensic tools to carry out the investigative process. This study aims to determine the forensic tools that are useful today and in the future.

5 Conclusions and Future Scope

TETRAD's cryptographic algorithm demonstrates strong performance in terms of encryption, decryption, and resource efficiency, making it a robust choice for enhancing data security. The algorithm offers fast encryption and decryption times, high throughput, and efficient memory usage, ensuring secure protection of sensitive information. TETRAD's low decryption times and high decryption throughput enable quick access to secured data, making it suitable for applications where swift data retrieval is crucial. The algorithm operates efficiently with limited system resources, making it advantageous for devices or systems with constrained memory. TETRAD's consistent performance across multiple test scenarios suggests its reliability in providing strong data security, making it a dependable choice for various applications. The combination of Foremost Tool and a hex editor offers a comprehensive solution for retrieving lost data, allowing for automated recovery of known file types and salvage of data from damaged or fragmented files.

In future, authors envision the continuous evolution and enhancement of TETRAD'S to meet the evolving challenges of data security in the dynamic cybersecurity landscape.

- Compatibility and Integration: Future works can focus on ensuring compatibility across platforms and integrating TETRAD'S with cloud security systems.
- Machine Learning Improvements: Exploring machine learning techniques can further enhance the performance and effectiveness of TETRAD'S cryptographic algorithm.
- Collaboration and Optimization: Collaborating with industry specialists can help optimize the efficiency of TETRAD'S and obtain certifications for widespread adoption.
- Awareness and Adoption: Efforts can be made to foster awareness about TETRAD'S and promote its widespread adoption in various applications where data security, speed, and resource efficiency are essential considerations.

References

1. Akanksha Mathur, “A Research paper: An ASCII value-based data encryption algorithm and its comparison with other symmetric data encryption algorithms”, International journal on Computer Science and Engineering (IJCSE). Vol. 4 No. 09. Pp. 1650-1657, September 2012.
2. Atul Kahate, Cryptography and Network Security Second Edition.
3. Behrouz A. Forouzan: “Cryptography and Network security” McGraw Hill companies (special Indian edition, Science, 2011).
4. William Stallings “Cryptography and Network Security”, 3rd Edition, Prentice-Hall Inc., 2005.
5. M. Panda, “Performance analysis of encryption algorithms for security,” 2017, doi:10.1109/SCOPES.2016.7955835.
6. Behrouz A. Forouzan: “Cryptography and Network security” McGraw Hill companies (special Indian edition, Science,2011) .
7. Gupta and N. K. Walia, "Cryptography Algorithms: A Review, “INTERNATIONAL JOURNAL OF ENGINEERING DEVELOPMENT AND RESEARCH, vol. 2, no. 2, pp. 1667-1672,2014.
8. A. Afonso, B. Oliveira, J. Pereira, and J. Silva, "Forensic data recovery from solid-state drives: A systematic literature review," Digital Investigation, vol. 34, pp. 1-13, 2020.
9. L. Chen, H. Zhang, and Q. Wang, "Forensic Data Recovery Techniques in Virtualized Environments: A Survey," IEEE Access, vol. 8, pp. 113438-113450, 2020.
10. E. Jones and R. Smith, "Challenges in Mobile Device Forensics: A Comprehensive Review," Journal of Digital Forensics, Security and Law, vol. 13, no. 3, pp. 13-34, 2018.
11. H. Kim and Y. Lee, "Forensic Investigation of Cloud Storage Services: A Review," International Journal of Digital Crime and Forensics (IJDCF), vol. 11, no. 1, pp. 1-21, 2019.

12. Avanija, J., K. E. Kumar, Ch Usha Kumari, G. Naga Jyothi, K. Srujan Raju, and K. Reddy Madhavi. "Enhancing Network Forensic and Deep Learning Mechanism for Internet of Things Networks." (2023).P. Dibb and M. Hammoudeh, "Forensic data recovery from android os devices: An opensource toolkit," Proc. - 2013 Eur. Intell. Secur.Informatics Conf. EISIC 2013, no. May, p. 226, 2013, doi:10.1109/EISIC.2013.58.
13. 2. S. Rao Polamuri, L. Nalla, A. D. Madhuri, S. Kalagara, B. Subrahmanyam and P. B. L. Aparna, "Analyse The Energy Consumption by Integrating the IOT and Pattern Recognition Technique," 2024 2nd International Conference on Disruptive Technologies (ICDT), Greater Noida, India, 2024, pp. 607-610, doi: 10.1109/ICDT61202.2024.10489265
14. G. S. Cho, "NTFS Directory Index Analysis for Computer Forensics," Proc. 2015 9th Int. Conf. Innov. Mob. InternetServ. Ubiquitous Comput. IMIS 2015, pp. 441-446, 2015