https://doi.org/10.48047/AFJBS.6.12.2024.4945-4956

African Journal of Biological Sciences

Journal homepage: http://www.afjbs.com

Research Paper Open Access

Comparative Analysis of Data Encryption Techniques in Cloud Environments

Suneetha Bandeela [1], S. Srithar [2], R.S.S. Bharadwaj [3], J. Sri Mokshagna [4], Abdul Afroz [5], P. Jeevan Sravanth [6]

[1] *Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Guntur, Andhra Pradesh, India. suneethab@kluniversity.in*

[2] *Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Guntur, Andhra Pradesh, India. sss.srithar@gmail.com*

[3] *Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Guntur, Andhra Pradesh, India. 2100030455cseh@gmail.com*

[4] *Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Guntur, Andhra Pradesh, India. 2100031166cseh@gmail.com*

[5] *Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Guntur, Andhra Pradesh, India. 2100030001cseh@gmail.com*

[6] *Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Guntur, Andhra Pradesh, India. 2100030402cseh@gmail.com*

Article History
Volume 6, Issue 12, 2024
Received: June 10, 2024
Accepted: July 5, 2024
doi:
10.48047/AFJBS.6.12.2024.4945-4956

Abstract-Maintaining timely and secure data transmission and processing in cloud computing environments requires effective data encryption and decryption procedures. Unfortunately, attaining the best encryption/decryption speed is frequently difficult for existing encryption techniques, which results in latency problems and performance bottlenecks. Innovative techniques and optimization strategies can maximize encryption/decryption speed while maintaining strong security measures by addressing resource constraints, algorithm complexity, and scalability limitations. Cloud computing platforms can process data more quickly, have lower latency, and provide better user experiences when encryption and decryption speeds are increased. This boosts the platform's capabilities and competitiveness. To cope with the growing demand for performance and data security.

Keywords: Data Encryption Decryption Algorithms, Cloud Security, Latency approximation, Cloud Computing, AES, QoS, Re-Encryption.

1. Introduction

1.1. Background

Offering unmatched scalability, flexibility, and accessibility, cloud computing has completely changed the data processing and storage landscape. The necessity of protecting private data against online attacks and illegal access however, grows as cloud-based services become more and more prevalent. In cloud environments, encryption is a vital component of data security, guaranteeing that information is shielded from capture and misuse.

1.2. Problem Description

The need to attain a security and performance balance often limits the effectiveness of encryption techniques, despite their critical role in protecting the confidentiality of data. The efficiency and usability of cloud-based applications are largely dependent on several factors, one of which is the speed at which encryption and decryption happen. In addition to reducing data transmission and processing speeds and degrading user experience, slow encryption and decryption processes can also introduce latency.

The context for discussing the significance of speeding up encryption and decryption in cloud environments is established by this introduction. We examine the difficulties that present encryption methods have in reaching the fastest possible speeds, such as scalability issues, algorithmic complexity, and resource limitations. The comparative analysis we carried out, points out the impact of taking encryption/decryption speed and scalability into consideration in cloud computing environments as a way to decrease latency, improve system efficiency.

1.3. Research Purpose

This study explores four distinct data encryption and decryption algorithms in cloud computing. By comparing their efficiency and applicability, the study seeks to determine which algorithm suits specific contexts best. The focus extends to understanding how faster encryption and decryption can enhance features and competitiveness in cloud platforms.

The objective is to offer thoughts into the present condition of the encryption/decryption speed in cloud environments, guiding future research and development efforts in data security. This study aims to offer a straightforward understanding to assist practitioners and researchers in making informed algorithmic choices.

2. Literature Review

Arora, R. et al. [1] Cloud computing has transformed how data is stored and accessed, providing scalability, flexibility, and cost savings. In cloud computing, encryption algorithms are vital for protecting sensitive user data. These algorithms ensure that the data is transformed into a form that is unintelligible for unauthorized parties. Several encryption algorithms have been proposed and implemented to safeguard data in cloud computing environments.

Dixit et al. [2] says Emerging concepts like compression- then-encryption and quantum encryption hold promise for improving security while reducing computational complexity. Future directions aim to create more efficient algorithms that combine compression and encryption with quantum concepts to withstand modern cryptanalysis.

Cao, H. et al. [6] demonstrates the importance of high- speed data encryption systems in education, particularly in fields such as traffic engineering, to ensure secure data transmission and privacy. Experimental results show that these systems improve data security and privacy in online education environments.

Malhotra, S. et al. [3] Says The significance of encryption in mitigating data security risks in cloud computing, among concerns about cyber-attacks and system errors. Symmetric Searchable Encryption (SSE) combined with Machine Learning, specifically Artificial Neural Networks (ANN), improves security and efficiency in cloud data storage and retrieval. Comparative studies show that encryption-based models outperform traditional classifiers, indicating the potential for improved data security and reliability in cloud environments.

Bhattacharjee, S. et al. [4] emphasizes the difficulties of ensuring integrity and privacy of data when using cloud data migration, with existing techniques such as encryption and steganography having limitations. These systems provide improved privacy, integrity, and efficiency in cloud data transmission, as demonstrated by comparative evaluations with traditional methods.

Mohammed, M.A. et al. [5] Highlights the importance of data security in cloud computing has prompted research into homomorphic encryption to guarantee data privacy and integrity.

Comparative testing demonstrates NAZUZ's higher efficiency in encrypting large file sizes, indicating that it holds promise for addressing advancing security challenges in cloud computing.

Amanipour et al. [7] says encryption has a significant impact on protecting cloud data warehousing by balancing security and performance trade-offs. Studies investigate encryption's applications, complexities, and consequences, advocating for informed decisions and ongoing security evaluation. A well-balanced approach is critical for improving data confidentiality, integrity, and availability in dynamic cloud environments.

Ahmad, S. et al. [8] identifies challenges to achieving optimal data security and quality of service (QoS) performance in cloud computing. Existing encryption techniques have constraints with respect to tamper resistance and quality of service (QoS). To address these issues, the suggested TLADE (Time-Oriented Latency Approximation - Based Data Encryption) algorithm selects optimal encryption schemes based on latency approximation and QoS values, thereby improving performance and lowering latency.

Rao, B.R. et al. [9] discusses the importance of cloud security and introduces an approach for protecting public clouds called HECC (Hybrid Elliptic Curve Cryptography). HECC optimizes AES encryption by using lightweight Edwards curves, Identity Based Encryption (IBE), and key reduction schemes. The performance evaluation demonstrates HECC's superiority in key creation time, throughput, and avalanche effect, emphasizing its potential for improving cloud data security and efficiency.

Krishnasamy, V. et al. [10] presents a novel approach to cloud storage security that employs the Index-level Boundary Pattern Convergent Encryption (IBPCE) and Array Preserving Triple Data Encryption (AP3DE)**.** These methods ensure data integrity, privacy, and efficiency while addressing flaws in current deduplication and encryption solutions. The proposed mechanisms provide strong protection against privacy breaches and brute-force attacks, improving the security of cloud storage systems.

K. Gai et al. [12] discusses the privacy issues raised by big data's rapid expansion in cloud computing. In order to maximize privacy protection while maintaining performance, it introduces a novel technique called Dynamic Data Encryption Strategy (D2ES), which focuses on selectively encrypting data within predetermined time constraints. All things considered, the study highlights how crucial scalable encryption methods are to maintaining confidentiality of data in cloud environments with growing data volumes.

Y. Yao et al. [13] presents searchable encryption schemes and lattice-based key-aggregate encryption, which have potential efficiency and resistance to quantum computing attacks. The plans are made to handle the difficulty of keeping track of encryption keys in cloud storage while maintaining effectiveness and security. In summary, the study highlights how crucial scalable encryption methods are for safely exchanging encrypted data in cloud storage.

K. Lee et al. [16] explains the important role of sharing data safely while making use of cloud storage. Although it doesn't specifically address scalable encryption, scalable techniques are necessary to manage the enormous datasets kept on cloud servers in an effective manner. This guarantees overall security, low cost, and quick performance.

Naik, R.M. et al. [19] describes a novel method that demonstrates quick encryption and decryption times for cloud data security and access control by combining PRE and CP- ABE. It highlights the model's resistance to collusion attacks and its suitability for devices with limited resources.

3. Methodology

3.1. Time-Oriented Latency Approximation-Based Data Encryption (TLADE)

The Time-Oriented Latency Approximation-Based Data Encryption (TLADE) offers a complete approach that highlights a novel data encryption algorithm to solve the critical

demand for enhanced data security within cloud storage. This section elaborates on the details of TLADE, offering a thorough comprehension of its functional architecture and the intricate operations of involved algorithms.

3.1.1 Working of the Preprocessing Algorithm

This phase requires the application of an extensive preprocessing technique. This algorithm's job is to refine the access trace set by efficiently eliminating unnecessary noise, which guarantees accurate route identification [8]. It examines each trace for the existence of any critical elements, then reveals a set of paths that might be used to transfer data in an efficient manner.

3.1.2 Approximation of Latency

Based on the routes that have been found, the approach carries out a critical step of approximating latency. Route discovery is the process of gathering detailed traffic information by carefully recording the details of every hop and router that are located in between the specified service location and the source. As an essential component, this traffic data is kept on the server for later analysis and well- informed decision-making.

Traffic-Throughput-Latency QoSV Estimation

Estimation of QoSV Throughput-Latency Traffic Latency Algorithm is at the top in the complex game of latency approximation [8]. This algorithm manages the calculation of support values for Quality of Service (QoS) along many routes. The algorithm provides QoS support values by analyzing latency, throughput, and traffic metrics in detail, which provides a detailed picture of the overall service quality of the cloud system.

3.1.3 QoS Support Measurement

The methodology keeps going in the same direction by measuring QoS support with precision. Preprocessing of access traces makes it possible to estimate QoS support values for specified routes. This complex analysis, performed at multiple time points, explores the dynamic interaction of variables like traffic, throughput, and latency. A detailed understanding of the quality of service at various points in time is the final outcome.

3.1.4 Data Encryption Scheme Selection

After setting the groundwork for route identification and QoS support assessment, TLADE focuses on data encryption techniques [8]. The mean delay of a carefully selected set of encryption algorithms is determined from the trace. By determining latency support based on data size and calculated latency, the method goes beyond surface-level assessment and assists in selecting a wise choice of encryption scheme that meets contextual requirements.

3.1.5 Optimal Route and Encryption Scheme Selection

The TLADE process concludes in the deliberate pairing of the best routes and encryption algorithms. By utilizing a variety of QoS support values and latency estimations, the methodology coordinates the process of choosing the best route for communicating data [8]. In order to guarantee a private and safe exchange, an encryption technique that is sensitive to the details of the data and its transmission requirements is selected at the same time. The technique considers the amount of time and the quantity of encrypted bytes, making sure that the best encryption strategy is carefully chosen to protect data integrity and confidentiality.

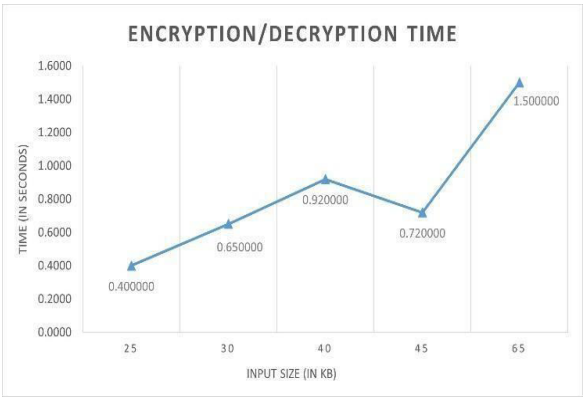


Figure 1. TLADE Encryption/Decryption Time vs Input Size

The graph provides insights into the performance of the TLADE (Time-Oriented Latency-Based Secure Data Encryption) algorithm concerning how much time it takes to encrypt and how much time it takes to decrypt. The execution time grows in proportion to the input size. Initially, for smaller input sizes (around 25 KB), the encryption/decryption time remains relatively low (approximately 0.4 seconds). However, at larger input sizes (65 KB), the time significantly escalates to 1.5 seconds.

3.2. Hybrid Elliptic Curve Cryptography (HECC)

An innovative cryptographic approach entitled HECC (Hybrid Elliptic Curve Cryptography) seeks to provide secure and rapid data encryption for security of public cloud. HECC seeks to optimize the encryption process while guaranteeing strong security measures by utilizing a mix of lightweight Edwards curves, IBE (Identity-Based Encryption), key compression methods, and the AES (Advanced Encryption Standard) [9].

3.2.1 Lightweight Edwards Curve for Key Generation

Lightweight Edwards curves are essential to key generation in the HECC model [9]. The mathematical idea known as Edwards curves is selected because it is effective in calculation and arithmetic operations. The author's investigation specifically employed the following equation: $p^2 + q^2 = 1 + 486662 * p^2 * p^2 \pmod{2255 - 19}$ [9]. Edwards curves are lightweight, which makes them ideal for effective key generation. It also makes them resilient to certain types of attacks and allows for quick arithmetic operations.

3.2.2 Identity-Based Encryption (IBE) for Key Modification

Identity-Based Encryption (IBE) introduces a unique approach by transforming user identities into public keys, enhancing system usability, and simplifying key management. This algorithm stands out from conventional encryption methods by prioritizing user authentication and authorization, a departure from the typical focus solely on data encryption and decryption. The effectiveness and security of this system are underscored by its ability to adapt private keys based on user IDs, providing a robust and user- centric solution for managing encryption keys.

3.2.3 Key Length Reduction

To optimize the encryption process a key compression approach is applied using Advanced Encryption Standard (AES). [9] By reducing the size of the private keys, this technique seeks to decrease computational overhead and increase encryption efficiency.

3.2.4 Elliptic Curve Diffie-Hellman (ECDH) Key Swap

Diffie-Hellman key swap protocol is used by Elliptic curve cryptography to create a shared secret between two parties. [9] This method protects confidentiality and enables the creation of a shared secret even when communication is occurring via an insecure channel.

3.2.5 Advanced Encryption Standard (AES) Encryption

The AES (Advanced Encryption Standard) is used by the HECC model to perform encryption. Strong security and effective encryption methods and decryption procedures are

characteristics of the extensively used and supported AES encryption technique. To provide strong encryption of the data, the encryption process includes multiple rounds of byte substitution, shift rows, mix columns, and the inclusion of round keys.

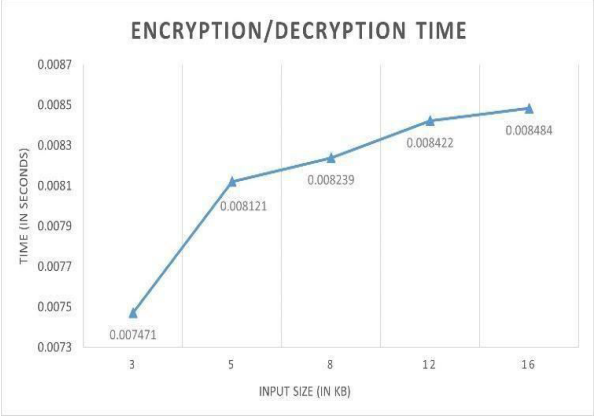


Figure 2. HECC Encryption/Decryption time vs Input Size

The graph provides valuable insights into the execution time of the Hybrid Elliptic Curve Cryptography (HECC) algorithm concerning how much time it takes to encrypt and decrypt a file. The execution time shows a positive association with increasing input size. For smaller input sizes (around 3 KB), the encryption/decryption time remains relatively low, approximately 0.0074 seconds. As the input size grows to 5 KB, the time increases slightly to 0.0081 seconds. At 8 KB, the algorithm's execution time further rises to 0.0082 seconds. Interestingly, there is a slight decrease at 12 KB, where the time drops to 0.0084 seconds. However, a significant increase occurs at the largest input size of 16 KB, reaching approximately 0.0085 seconds.

3.3. Index-level Boundary Pattern Convergent Encryption (IBPCE)

An innovative method for enhancing data security and reducing cloud storage expenses is the IBPCE (Index-level Boundary Pattern Convergent Encryption) [10]. Cloud data integrity and secrecy are guaranteed by IBPCE, which incorporates innovative techniques including Convergent Encryption (CE), index-level boundary analysis, and duplicate content identification.

2.2.1 Index-level Boundary Duplicate File Examination

In order to identify duplicate material, the system first examines files at the index level. Papers are converted to bytes, and the original information is compared to other papers stored on the cloud. Based on a predetermined threshold, it finds duplicates using a boundary matching strategy. It stops being kept in the cloud if a duplicate is discovered.

2.2.2 Convergent Encryption (CE)

The algorithm encrypts the data using Convergent Encryption (CE) after duplication analysis is finished and stores it in the cloud. Utilizing a 128-bit key for managing ciphertext blocks and 64-bit plaintext, CE is a sort of symmetric key encryption [10]. The steps in the encryption process are converting plaintext to ASCII code, making a square matrix, applying the key value via an XOR operation, and using the converted value-form Hexa code Transferred matrix. By doing this, safe encryption is ensured before the data is stored.

2.2.3 Distribution of Encrypted Records

The technique distributes the encrypted records among many servers after encryption. By ensuring the security and integrity of the data distributed over several servers, this approach reduces the possibility of data corruption and improves overall data protection. Additionally, this distribution facilitates effective data management and retrieval.

2.2.4 Authentication and Access Control

In order to guarantee that only authorized users may access and retrieve the encrypted information stored in the cloud, the technique includes an authentication role for cloud- based

users. The process of maintaining and accessing data is made more secure by an additional layer of authentication.

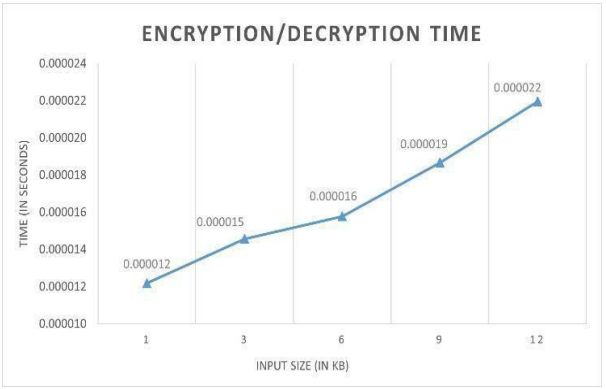


Figure 3. IBPCE Encryption/Decryption Time vs Input Size

The graph represents the relationship between input size (in kilobytes) and the time it takes to encrypt or decrypt that input (in seconds). The x-axis ranges from 1 to 12 KB, representing the input size, while the y-axis ranges from 0.000010 to 0.000024 seconds, representing the time. The graph shows a positive correlation, indicating that as the input size increases, the time for encryption or decryption also increases. This suggests that larger files will take longer to encrypt or decrypt. However, even for the largest input size of 12 KB, the time remains extremely small, indicating that the process is relatively fast. Specific data points are labeled at input sizes of 1, 3, 6, and 12 KB.

3.4. CP Attribute-Based Encryption with PRE

An innovative cryptographic approach called CP-ABE- PRE (Ciphertext-Policy Attribute-Based Encryption with Proxy Re-Encryption) is intended to enhance access control and data safety and access management in cloud computing. With the integration of Proxy Re-Encryption (PRE) and CipherText-Policy Attribute-Based Encryption (CP-ABE), CP-ABE-PRE provides safe data governance as well as effective key administration for cloud data encryption.

3.4.1 CP Attribute-Based Encryption (CP-ABE)

A cryptographic strategy entitled CP-ABE (Ciphertext- Policy Attribute-Based Encryption) effectively handles the keys required to encrypt cloud data. Every encrypted piece of data in CP-ABE is connected to a policy according to certain criteria. Material can only be decrypted by users if the properties of the private key they are employing correspond to the authorized access hierarchy linked to the encrypted information [19]. CP-ABE serves the purpose for safe data sharing and collaboration since it offers necessary choices for allocating and managing access to cloud data.

3.4.2 Proxy Re-Encryption (PRE)

Another cryptographic approach that promises safe and discreet cloud data exchange is proxy re-encryption. Without disclosing the plaintext to a middleman, it enables the conversion of ciphertexts encrypted with one user's public secret key into ciphertexts which can be decoded with another user's private secret key. PRE tackles privacy and security issues with cloud-based collaboration and data exchange.

3.4.3 Working of CP-ABE-PRE

3.4.3.1. Key Generation

The public secret key and master secret key are created by a Trusted Authority (TA) [19]. User attributes are given unique codes, which serve as a revision key in case of future cancellations.

3.4.3.2. Attribute-Based Key Generation

The public and private keys that are generated by attributes are used by the data owner to protect files that are transmitted to the cloud server.

3.4.3.3. Data Encryption

An intermediary decryption step that may reveal the plaintext is prevented by transforming the encrypted data in a way that makes it unlockable with fresh keys. Secure cloud data storage is ensured by this procedure, which is called data re-encryption.

3.4.3.4. Proxy Re-Encryption Strategy

The ABE (Attribute-Based Key Generation) technique uses the Proxy Re-Encryption mechanism when combined with a proxy server. This makes it possible for cloud data to be shared securely [19]. An authorized user may access the data by using their key to encrypt the ciphertext again and then decrypting the resulting file.

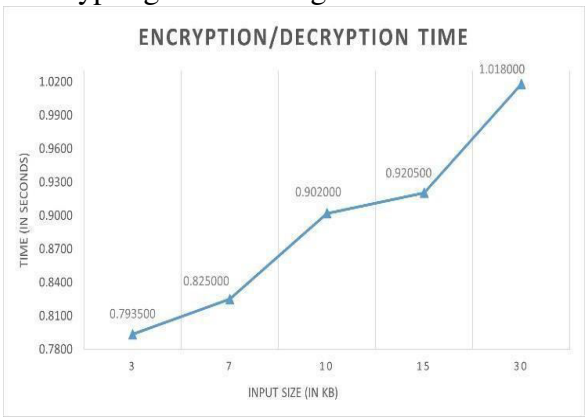


Figure 4. CP-ABE-PRE Encryption/ Decryption Time vs Input Size

The graph illustrates the relationship between input size (in kilobytes) and the time it takes to encrypt or decrypt that input (in seconds). The x-axis ranges from 3 to 30 KB, representing the input size, while the y-axis ranges from approximately 0.7800 to slightly over 1.0200 seconds, representing the time. The graph shows a nonlinear increase in time as the input size increases. For instance, at 3 KB, it takes approximately 0.793500 seconds; at 7 KB, it takes about 0.825000 seconds; and at 30 KB, it takes roughly 1.018000 seconds. This suggests that larger files will take longer to encrypt or decrypt. However, even for the largest input size of 30 KB, the time remains under 1.02 seconds, indicating that the process is relatively fast.

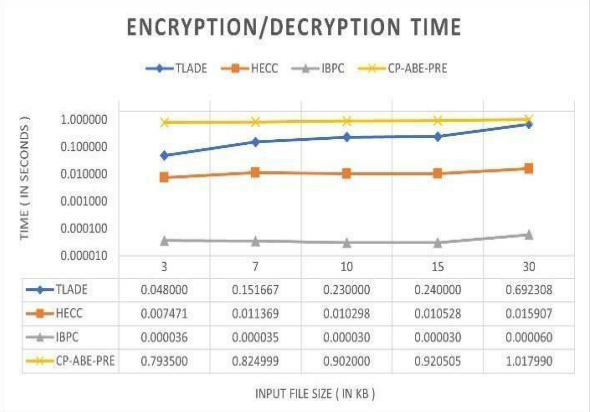
Table I. Key Steps for Techniques

Techniques	Key Steps
TLADE (Time-Oriented Latency Approximation-Based Data Encryption)	Identifying the possible routes and services, Finding or Estimating the Latency, Quality of service measures, Finding the optimal route, and Selecting Which scheme for encryption.
HECC (Hybrid Elliptic Curve Cryptography)	Generating key using Lightweight Edwards Curve, transforming user identities into public keys, compressing key length, Creating a shared secret, and Performing encryption using AES.

IBPCE (Index-level Boundary Pattern Convergent Encryption)	Finds duplicates using a boundary matching strategy, Performing symmetric key encryption, Distribution of Encrypted Records, and Authentication and Access Control.
CP-ABE-PRE (Ciphertext-Policy Attribute-Based Encryption with Proxy Re-Encryption)	Generation of public secret and master secret key, Data owner generating private key, Encryption of data, and Data access using keys by re-encryption of cyphertext.

4. Result and Experimentation

Based on Fig1, Fig2, Fig3 and Fig4, the below analysis is generated by taking the ratios of encryption/decryption time and file size for each graph.



	3	7	10	15	30
TLADE	0.048000	0.151667	0.230000	0.240000	0.692308
HECC	0.007471	0.011369	0.010298	0.010528	0.015907
IBPC	0.000036	0.000035	0.000030	0.000030	0.000060
CP-ABE-PRE	0.793500	0.824999	0.902000	0.920505	1.017990

Figure 5. Comparing all the Above-mentioned Techniques

The graph illustrates the encryption/decryption time of four different algorithms (TLADE, HECC, IBPC, CP-ABE- PRE) as the input file size increases. The y-axis is a logarithmic scale representing time in seconds, and the x-axis represents the input file size in KB.

4.1.Comparison

By providing a comparative analysis, the writers can point out any similarities or differences between their findings and earlier studies on the same topic. A greater comprehension of the outcomes of their conclusions is made possible by these comparisons. As they connect to previous research, their findings take on a greater significance, highlighting the applicability and value of their contributions to the field. Examining these similarities and differences strengthens the background and establishes the importance of their work in relation to the study's main theme.

4.2.TLADE (Time-Oriented Latency Approximation-Based Data Encryption)

TLADE shows a significant increase in time as the input file size increases, indicating that it may not be as efficient as some of the other algorithms when dealing with larger files. Compared to HECC and IBPC, TLADE takes considerably more time. However, it is faster than CP-ABE-PRE, especially for larger file sizes. It might be because of increased processing overhead for route identification, latency approximation, and QoS measurement. Explore lightweight methods for route discovery and QoS estimation to reduce processing overhead and to increase efficiency.

4.3.HECC (Hybrid Elliptic Curve Cryptography)

HECC also shows an increase in time with increasing file size, but it is less than that of TLADE and CP-ABE-PRE, making it more efficient than these two. However, compared to IBPC, HECC takes more time, indicating that IBPC may be more efficient for larger file sizes. There can be potential security trade-offs compared to other algorithms depending on specific implementation choices. Rigorous security analysis is needed to ensure the chosen parameterization and key lengths offer adequate security guarantees.

4.4. IBPCE (Index-level Boundary Pattern Convergent Encryption)

IBPC stands out as it shows almost no increase in time, regardless of the input file size. This suggests that IBPC is highly efficient, outperforming TLADE, HECC, and CP- ABE-PRE in terms of speed, especially for larger file sizes. But it relies on the effectiveness of Convergent Encryption, which might not be suitable for highly sensitive data, and additional processing power required for duplicate detection. Investigate alternative encryption schemes with stronger security guarantees for sensitive data. Explore techniques for optimizing duplicate detection to minimize processing overhead.

4.5. CP-ABE-PRE (Cipher Text-Policy Attribute-Based Encryption with Proxy Re-Encryption)

CP-ABE-PRE has the highest time among the four algorithms, especially for larger file sizes, indicating that it is the least efficient in terms of speed. When compared to TLADE, HECC, and IBPC, CP-ABE-PRE takes the most time. It has higher computational complexity compared to other algorithms due to attribute-based encryption and re- encryption overhead. It also requires proper infrastructure for managing user attributes and policies, which can be complex to implement. A user-friendly and efficient tool for managing user attributes and policies to simplify implementation.

Table II. Comparison of the Techniques

INPUT FILE SIZE (KB)	ENCRYPTION / DECRYPTION TIME			
	TLADE (S)	HECC (S)	IBPC (S)	CP-ABE-PRE (S)
3	0.048000	0.007471	0.000036	0.793500
7	0.151667	0.011369	0.000035	0.824999
10	0.230000	0.010298	0.000030	0.902000
15	0.240000	0.010528	0.000030	0.920505
30	0.692308	0.015907	0.000060	1.017990

5. Conclusion

The research explores four distinct data encryption algorithms TLADE (Time-Oriented Latency Approximation - Based Data Encryption), HECC (Hybrid Elliptic Curve Cryptography), IBPCE (Index-level Boundary Pattern Convergent Encryption), and CP-ABE-PRE (Ciphertext - Policy Attribute-Based Encryption with Proxy Re - Encryption) aiming to determine best suitable context and priorities in which these techniques can be applicable to enhance the data safety in cloud computing.

When prioritizing low latency and optimal performance, the HECC (Hybrid Elliptic Curve Cryptography) algorithm emerges as a strong contender.

In cases where storage efficiency takes precedence, the IBPCE (Index-level Boundary Pattern Convergent Encryption) algorithm proves to be a fitting choice.

For situations demanding meticulous access control and secure data sharing, the CP-ABE-PRE (Ciphertext-Policy Attribute-Based Encryption with Proxy Re-Encryption) algorithm stands out as the optimal solution.

When seeking a cohesive approach that strikes a balance between performance, security, and Quality of Service (QoS), the TLADE (Time-Oriented Latency Approximation - Based Data Encryption) algorithm shows itself as the most suitable option. performance, security, and QoS is desired, TLADE might be the most fitting option.

References

[1] Arora, R., Parashar, A. and Transforming, C.C.I., 2013. Secure user data in cloud computing using encryption algorithms. International journal of engineering research and applications, 3(4), pp.1922-1926.

[2] Dixit, Pooja & Gupta, Avadhesh & Trivedi, (Dr.) Munesh & Yadav, Virendra. (2018). Traditional and Hybrid Encryption Techniques: A Survey. 10.1007/978-981-10-4600-1_22.

[3] Malhotra, S. and Singh, W., 2023. An efficacy analysis of data encryption architecture for cloud platform. Procedia Computer Science, 218, pp.989-1002.

[4] Bhattacharjee, S., Sharma, H., Choudhury, T. and Abdelmoniem, A.M., 2024. Leveraging chaos for enhancing encryption and compression in large cloud data transfers. The Journal of Supercomputing, pp.1-35.

[5] Mohammed, M.A. and Al Attar, T.N.A., 2023. Fully Homomorphic Encryption Scheme for Securing Cloud Data. UHD Journal of Science & Technology, 7(2).

[6] Cao, H. and Srivastava, G., 2023. An encryption transmission system for high-speed private data streams in online education in the specialty of "Traffic Engineering". Mobile Networks and Applications, pp.1-12.

[7] Amanipour, Soheil & Klaus, Hubert. (2024). Role of Encryption in Cloud Data Warehousing: Trade-offs between security and performance in encrypted cloud data warehouses.

[8] Ahmad, S. and Mehfuz, S., 2024. Efficient time-oriented latency-based secure data encryption for cloud storage. Cyber Security and Applications, 2, p.100027.

[9] Rao, B.R. and Sujatha, B., 2023. A hybrid elliptic curve cryptography (HECC) technique for fast encryption of data for public cloud security. Measurement: Sensors, 29, p.100870.

[10] Krishnasamy, V. and Venkatachalam, S., 2023. An efficient data flow material model based cloud authentication data security and reduce a cloud storage cost using Index-level Boundary Pattern Convergent Encryption algorithm. Materials Today: Proceedings, 81, pp.931-936.

[11] H. Deng et al., "Identity-Based Encryption Transformation for Flexible Sharing of Encrypted Data in Public Cloud," in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 3168-3180, 2020.
doi: 10.1109/TIFS.2020.2985532.

[12] K. Gai, M. Qiu and H. Zhao, "Privacy-Preserving Data Encryption Strategy for Big Data in Mobile Cloud Computing," in IEEE Transactions on Big Data, vol. 7, no. 4, pp. 678-688, 1 Oct. 2021.
doi: 10.1109/TBDATA.2017.2705807.

[13] Y. Yao, Z. Zhai, J. Liu and Z. Li, "Lattice-Based Key-Aggregate (Searchable) Encryption in Cloud Storage," in IEEE Access, vol. 7, pp. 164544-164555, 2019.
doi: 10.1109/ACCESS.2019.2952163.

[14] X. Huang, H. Xiong, J. Chen and M. Yang, "Efficient Revocable Storage Attribute-based Encryption With Arithmetic Span Programs in Cloud-Assisted Internet of Things," in IEEE Transactions on Cloud Computing, vol. 11, no. 2, pp. 1273-1285, 1 April-June 2023.
doi: 10.1109/TCC.2021.3131686.

[15] Sajay, K.R., Babu, S.S. and Vijayalakshmi, Y., 2019. Enhancing the security of cloud data using hybrid encryption algorithm. Journal of Ambient Intelligence and Humanized Computing, pp.1-10.

[16] K. Lee, "Comments on “Secure Data Sharing in Cloud Computing Using Revocable-Storage Identity-Based Encryption”," in IEEE Transactions on Cloud Computing, vol. 8, no. 4, pp. 1299-1300, 1 Oct.- Dec. 2020, doi: 10.1109/TCC.2020.2973623.

[17] L. Jiang and D. Guo, "Dynamic Encrypted Data Sharing Scheme Based on Conditional Proxy Broadcast Re-Encryption for Cloud Storage," in IEEE Access, vol. 5, pp. 13336-13345, 2017.
doi: 10.1109/ACCESS.2017.2726584.

[18] El-etriby, S., Mohamed, E.M. and Abdul-kader, H.S., 2012, March. Modern encryption techniques for cloud computing. In ICCIT (pp. 800-805).

[19] Naik, R.M., Gadiyar, H.M.T., Kumar, M.B., Jeevitha, B.K., Thyagaraju, G.S., Ujwal, U.J., Arjun, K., Manasa, S.M., Avinash, S., Kumar, J.A. and Sowmya, T.K., 2023. Original Research Article Key management and access control based on combination of cipher text - policy attribute-based encryption with Proxy Re-Encryption for cloud data. Journal of Autonomous Intelligence, 6(3).