ISSN 2277-2685
IJESR/July-September. 2026/ Vol-16/Issue-3/337-346
Mission Franklin *et. al.*, / International Journal of Engineering & Science Research
Full Length Article

Comparative Analysis of Database Encryption Techniques and Their Performance Overheads

Mission Franklin[1], Kufre Christopher Ukpe[2]
[1]Department of Computer Engineering, Rivers State University, Port Harcourt, Rivers State, Nigeria
[2]Department of computer Science, Obong university, Obong Ntak, Akwa Ibom State, Nigeria.
Email: mission.franklin@ust.edu.ng[1], ukpekaycee123@gmail.com[2]
Accepted 13-09-2026

Abstract
Database encryption is essential for protecting sensitive information, maintaining data confidentiality, and meeting regulatory requirements in modern data-driven applications. However, encryption can introduce computational overhead that affects query performance, resource consumption, and system scalability. Common approaches, including Transparent Data Encryption (TDE), column-level encryption, and client-side encryption, provide different levels of protection and performance trade-offs. Despite their widespread use, limited empirical evidence exists on how these techniques perform under comparable transactional and analytical workloads. This study presents a comparative evaluation of major database encryption strategies based on key performance indicators, including query latency, throughput, CPU and memory utilization, storage overhead, and index efficiency. A reproducible experimental framework is used to assess the performance implications of each encryption approach under controlled workloads. The study examines how encryption-related processing overhead influences database responsiveness and resource utilization while considering the security benefits provided by each technique. The findings are expected to provide practical evidence of the performance-security trade-offs associated with database encryption and support informed decisions by database administrators, developers, and system designers. The study contributes to the development of database security strategies that achieve an appropriate balance between strong data protection, query performance, and operational efficiency.

Keywords: *Database Encryption, Transparent Data Encryption, Column-level Encryption, Client-side Encryption, Query Performance, Data Security, Security–Performance Trade-offs*

1. Introduction

The rapid growth of digital technologies and cloud-based information systems has significantly increased the volume of sensitive data stored in Database Management Systems (DBMSs). As organizations increasingly depend on databases for managing financial records, healthcare information, academic records, customer transactions, and government data, protecting stored information from unauthorized access has become a major cybersecurity priority. The increasing frequency of cyberattacks, ransomware incidents, insider threats, and data breaches has further intensified the need for stronger database security mechanisms. In addition, global data protection regulations such as the General Data Protection Regulation (GDPR) (EP-CEU, 2016), the Nigeria Data Protection Act (NDPR Act 2023)(FRN, 2023), the Health Insurance Portability and Accountability Act (HIPAA)(USDHHS, 2013), and the Payment Card Industry Data Security Standard (PCI-DSS)(PCI-SSC, 2022) require organizations to implement robust security measures to safeguard sensitive information (EP-CEU, 2016 ; FRN, 2023; Oracle, 2025). Consequently, database encryption has emerged as one of the most effective approaches for ensuring data confidentiality and privacy in modern information systems.

Database encryption refers to the process of converting plaintext data into unreadable ciphertext using cryptographic algorithms, thereby preventing unauthorized users from interpreting stored information. Several encryption strategies are commonly implemented in modern DBMS environments. Transparent Data Encryption (TDE) protects data at rest by automatically encrypting database files without requiring modifications to applications. Column-level encryption secures specific sensitive attributes such as passwords, credit card numbers, and medical records, while client-side encryption ensures that data is encrypted before transmission to the database server (Mattsson, 2005)

These encryption approaches frequently utilize cryptographic standards such as the Advanced Encryption Standard (AES), particularly AES-256, because of its high level of security and computational efficiency (Carvalho et al., 2025; Microsoft, 2025b). Modern cloud database platforms including Microsoft SQL Server, Oracle

Database, PostgreSQL, and MySQL increasingly integrate encryption technologies as core security features to support compliance and protect sensitive enterprise data (Oracle, 2025; PostgreSQLGDG, 2025).

Despite the security advantages offered by database encryption, the implementation of encryption mechanisms often introduces performance overheads that may affect overall database efficiency. Encryption and decryption operations require additional computational processing, which increases CPU utilization, memory consumption, storage overhead, and disk input/output activities. Encryption can also reduce indexing efficiency, alter execution plans, and negatively impact query response times, especially in systems supporting high-volume transactional processing and complex analytical workloads (Vidaković et al., 2024).

According to Microsoft (2025b), enabling Transparent Data Encryption may slightly affect transaction throughput and backup operations due to the additional cryptographic processing involved. Similarly, Oracle (2025) noted that encrypted databases may experience increased latency during read and write operations, particularly when encrypted columns are heavily utilized in query execution.

Although previous studies have explored encryption-related performance issues, existing literature still lacks comprehensive comparative investigations covering multiple encryption techniques and workload categories within unified benchmarking environments. Many earlier studies focus primarily on single DBMS platforms or isolated encryption methods, thereby limiting the availability of generalized empirical evidence for database administrators and system designers (Ahsan et al., 2025). Furthermore, there remains insufficient understanding of how different encryption approaches influence transactional workloads compared with analytical workloads, as well as how selective encryption techniques compare with full-database encryption in terms of operational efficiency and resource utilization (Carvalho et al., 2025; Vidaković et al., 2024).

This study addresses these limitations by conducting a comparative performance evaluation of major database encryption techniques, including Transparent Data Encryption, column-level encryption, and client-side encryption, under realistic transactional and analytical database workloads. The study examines critical performance metrics such as query latency, throughput, CPU utilization, memory consumption, and storage overhead in order to quantify the operational impact associated with different encryption strategies.

The study contributes to the existing body of knowledge in four major ways. First, it provides a systematic comparison of widely adopted database encryption techniques against a non-encrypted baseline environment. Second, it introduces a reproducible experimental benchmarking framework for evaluating encrypted database systems. Third, it presents empirical evidence regarding the performance implications of encryption across diverse workload scenarios. Finally, the study offers practical recommendations that can assist organizations, database administrators, and system architects in selecting encryption approaches that align with operational requirements, compliance obligations, and security objectives.

1.1 Problem Statement

The increasing use of database systems for storing sensitive and mission-critical information has made data security a major concern for organizations. Rising cyber threats, data breaches, and regulatory requirements such as GDPR, NDPA and HIPAA have encouraged organizations to adopt database encryption techniques such as Transparent Data Encryption (TDE), column-level encryption, and client-side encryption to protect sensitive data (Carvalho et al., 2025). Although these encryption methods improve confidentiality and reduce unauthorized access, they also introduce performance overheads that may affect query execution time, throughput, CPU utilization, memory usage, and storage efficiency, particularly in transactional and analytical database environments (Vidaković et al., 2024).

Despite the widespread adoption of database encryption, there is still limited comparative research on the performance impact of different encryption techniques under realistic workloads. Most existing studies focus on individual encryption methods or theoretical security benefits, providing limited practical guidance for organizations seeking to balance security and database performance. In particular, the trade-offs between full-database encryption and selective encryption approaches have not been sufficiently evaluated.

Therefore, this study conducts a comparative performance analysis of major database encryption techniques to evaluate their impact on database efficiency under different workloads. The findings will provide practical insights for selecting encryption strategies that balance security requirements with operational performance.

1.2 Aim and Objectives of the Study

This study aims to compare the performance impact of major database encryption techniques under realistic workloads, with the goal of balancing data security and system efficiency.

The study objectives include:

1. Review key encryption methods: TDE, column-level, and client-side encryption.
2. Design a benchmark framework for OLTP and OLAP workloads.
3. Measure performance using latency, throughput, CPU/memory usage, and storage overhead.
4. Compare security and performance trade-offs across encryption techniques.
5. Provide practical guidance for selecting suitable encryption strategies.

1.3 Significance of the Study

This study provides practical insights into balancing data security and system performance in database environments. By comparing Transparent Data Encryption (TDE), column-level encryption, and client-side encryption, it highlights the trade-offs between security, latency, throughput, and resource usage. The findings will help database administrators and system designers select suitable encryption strategies while meeting regulatory requirements such as GDPR, NDPR, HIPAA, and PCI-DSS. The study also contributes to existing knowledge by offering a comparative evaluation of encryption techniques and provides a basis for future research on advanced database and cloud security solutions.

1.4 Scope of the Study

This study compares three database encryption techniques, such as Transparent Data Encryption (TDE), column-level encryption, and client-side encryption, and their impact on relational database performance. The study evaluates OLTP and OLAP workloads using metrics such as query latency, throughput, CPU/memory usage, and storage overhead. The study focuses on performance trade-offs, excluding NoSQL databases and advanced encryption methods such as homomorphic encryption.

2. Literature Review

2.1 Conceptual Review

1) **Concept of Database Encryption:** Database encryption is the process of converting readable data (plaintext) into an unreadable format (ciphertext) to prevent unauthorized access. It is a core security mechanism in modern database systems designed to protect sensitive information both at rest and during storage operations. In contemporary DBMS environments, encryption is widely adopted to support regulatory compliance and protect against data breaches, especially in cloud and enterprise systems (Carvalho et al., 2025). The main idea is that even if attackers gain access to database files, the data remains unusable without the correct cryptographic keys.
2) **Objectives of Database Encryption:** The primary objectives of database encryption are confidentiality, integrity, and regulatory compliance. Confidentiality ensures that only authorized users can access data, while integrity guarantees that data is not altered without permission. Compliance is equally important, as frameworks such as GDPR, NDPA, HIPAA, and PCI-DSS require organizations to implement strong data protection mechanisms. Modern encryption systems also aim to ensure secure data storage without disrupting database functionality (Oracle, 2025; Sharma et al., 2024).).
3) **Types of Database Encryption Techniques**
   a) ***Transparent Data Encryption (TDE):*** TDE encrypts entire database files automatically at the storage level without requiring changes to application code. It encrypts data before writing it to disk and decrypts it when reading into memory. This makes it easy to implement in enterprise systems. However, studies show that it may introduce overhead during I/O operations and backups due to continuous encryption and decryption processes (Microsoft, 2025; Oracle, 2025).
   b) ***Column-Level Encryption:*** Column-level encryption secures specific fields within a table rather than the entire database. This approach is commonly used for highly sensitive attributes such as passwords and financial data. It offers better flexibility but may reduce query efficiency because encrypted fields are harder to index and process (Microsoft, 2025b).
   c) ***Client-Side Encryption:*** Client-side encryption ensures that data is encrypted before it reaches the database server. This means the server only stores ciphertext and cannot access plaintext data. While this provides strong security, it limits server-side processing capabilities such as searching, sorting, and indexing, shifting more workload to the application layer (Oracle, 2025).
4) **Mechanism of Database Encryption in DBMS:** Database encryption operates using cryptographic keys and algorithms, often managed through a hierarchical key system. For example, TDE uses a database encryption key (DEK) protected by certificates or master keys. Data is encrypted during write operations and decrypted during read operations. This process is typically transparent to users and applications, ensuring security without modifying business logic (Microsoft, 2025b).
5) **Role of Cryptographic Algorithms in Database Security:** Most modern DBMS

encryption systems rely on strong algorithms such as Advanced Encryption Standard (AES-256). AES is widely used because it provides high security while maintaining acceptable performance levels (Kim et al., 2025). Hardware acceleration technologies such as Advanced Encryption Standard New Instruction (AES-NI) further reduce encryption overhead, making it suitable for large-scale database environments (NIST, 2023; Carvalho et al., 2025).

6) **Security Benefits of Database Encryption:** Database encryption significantly improves data protection by preventing unauthorized access, reducing risks from stolen storage media, and protecting sensitive information in cloud environments. It also supports compliance with international data protection regulations. Additionally, encryption helps maintain trust between organizations and users by ensuring that personal and financial data is securely handled (Oracle, 2025).
7) **Performance Implications of Database Encryption:** Although encryption enhances security, it introduces performance overhead. Encrypted databases often experience increased CPU usage, higher memory consumption, and longer query execution times. Research shows that encryption can also slow down read/write operations and backup processes, depending on workload intensity and system configuration (Carvalho et al., 2025; Microsoft, 2025).
8) **Encryption and Database Performance Trade-offs:** A key issue in database encryption is the trade-off between security and performance. Stronger encryption improves data protection but may reduce system efficiency. For example, TDE offers ease of use but affects I/O performance, while column-level encryption improves control but reduces query speed. Organizations must balance these factors based on operational needs and security requirements (Popa et al., 2011).
9) **Impact of Encryption on Query Processing and Indexing:** Encryption affects how databases process queries and use indexes. Since encrypted data is transformed into ciphertext, it does not maintain natural ordering, making indexing less efficient. This can increase query execution time, especially in search-heavy systems. As a result, query optimization becomes more complex in encrypted environments (Vidaković et al., 2024).
10) **Workload Sensitivity in Encrypted Databases (OLTP vs OLAP):** The performance impact of encryption varies depending on workload type. OLTP systems, which handle frequent small transactions, are more sensitive to latency and may experience noticeable slowdowns under encryption. OLAP systems, which process large analytical queries, may experience higher resource consumption and longer processing times. This makes workload type a critical factor in evaluating encryption performance (Carvalho et al., 2025; Lee et al., 2019).).

2.2 Empirical Review

Empirical studies across different researchers consistently show that database encryption improves security but introduces measurable performance overheads in relational and cloud database environments.

- Carvalho et al. (2025) conducted a comparative analysis of AES-based encryption across Oracle, MySQL, and Microsoft SQL Server and found that encrypted databases consistently exhibited slower query execution times compared to plaintext database configurations. Their findings also revealed that the level of performance degradation varied across platforms depending on workload intensity and encryption implementation.
- Vidaković et al. (2024) reported that database encryption with AES-256-GCM (Advanced Encryption Standard 256-bit in Galois/Counter Mode)(NIST, (2007) in web-based applications leads to higher response times, particularly under concurrent user access. The study highlights that encryption overhead becomes more significant in multi-user environments where frequent transactions occur, thereby affecting system scalability and responsiveness.
- Alenezi et al. (2024) examined encrypted cloud database systems and found that encryption negatively affects scalability and storage efficiency in distributed environments, especially when handling large datasets. The authors result suggest that encryption overhead increases with data volume and system complexity.
- Khan et al. (2022) conducted an analytical evaluation of encrypted query processing in Database-as-a-Service (DaaS) environments and concluded that encryption and decryption require substantial computational resources and introduce performance overhead, making query execution on encrypted databases more time-consuming. They further suggested that parallel query execution through multithreading can mitigate some of these performance limitations.
- Tawose et al. (2023) reported that the computational overhead associated with homomorphic encryption can degrade the performance of outsourced database systems during encrypted query processing. Their proposed parallel caching framework mitigates

this overhead by improving query execution efficiency and reducing processing time.

- Madyatmadja et al. (2021) found that Transparent Data Encryption (TDE) introduces moderate performance overhead in Microsoft SQL Server, resulting in increased CPU utilization, memory consumption, and backup duration, while reducing transaction throughput. Despite these performance costs, the authors concluded that TDE remains a practical mechanism for protecting data at rest. These findings are consistent with Microsoft and Oracle documentation, which indicates that encrypted database workloads may require additional CPU and I/O resources, particularly during backup and recovery operations.
- Sun et al. (2024) noted that query processing over encrypted databases presents significant optimization challenges because conventional indexing, sorting, and range-query operations are difficult to perform on encrypted data. The authors emphasized that selective encryption and specialized encrypted indexing techniques provide a practical balance between security and query performance.

Empirical studies consistently show that database encryption improves data confidentiality and regulatory compliance but introduces performance trade-offs in latency, throughput, CPU usage, memory consumption, and scalability. The impact varies depending on the encryption technique, workload, and system configuration, making it essential to balance security with performance. However, existing research mainly examines individual encryption methods or specific environments, with limited comparative evaluation of Transparent Data Encryption (TDE), Column-level encryption, and Client-side encryption under both OLTP and OLAP workloads. In addition, there is insufficient analysis of comprehensive performance metrics and limited practical guidance for selecting suitable encryption strategies in modern cloud, hybrid, and edge computing environments.

3. Methodology

This study adopts a quantitative experimental research design to evaluate and compare the performance overhead of major database encryption techniques under controlled conditions. The approach allows for objective measurement of system behaviour under different encryption configurations and workloads.

3.1 Research Design: The study is based on a comparative benchmarking design, where three encryption techniques: Transparent Data Encryption (TDE), Column-level encryption, and Client-side encryption are evaluated against a non-encrypted baseline. This design enables direct comparison of performance differences under identical experimental conditions.

3.2 Experimental Environment, Dataset and Workloads: The study was conducted using MySQL 8.0 deployed on a Windows 11 workstation with an Intel Core i7 processor, 16 GB RAM, and a 512 GB SSD. All experiments were performed under identical hardware and software configurations to ensure fair and reproducible results. Transparent Data Encryption, Column-level encryption, and Client-side encryption were implemented using the AES-256 encryption algorithm.

3.3 Performance Evaluation and Data Analysis: Database performance is evaluated using key metrics, including query latency, throughput (transactions per second), CPU and memory utilization, storage overhead, and index efficiency. Performance data are collected using built-in DBMS monitoring tools and system profiling utilities, with each experiment repeated multiple times to ensure reliability. The results are analysed using descriptive statistics, including mean values, percentage changes, and comparative analysis, and are presented in tables and charts to highlight performance differences among the encryption techniques.

3.4 Validity and Reliability: To ensure validity and reliability, all experiments are conducted under identical system conditions using standard encryption implementations, with repeated trials and consistent workload execution. This structured and reproducible methodology enables a fair comparison of database encryption techniques and their performance trade-offs across different workloads.

4. Results and Analysis

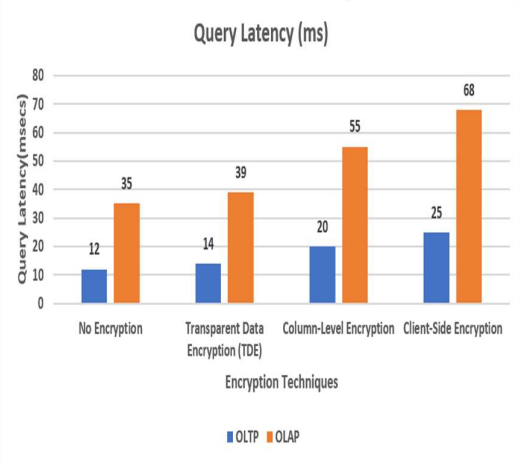

Figure 1 Query Latency (ms)

The query latency of the experimental comparison is shown in Figure 1, which shows that, query latency increases as stronger encryption techniques are applied. No Encryption achieved the lowest latency (12 ms for OLTP and 35 ms for OLAP), while Transparent Data Encryption (TDE) introduced only a small increase (14 ms and 39 ms). Column-Level Encryption resulted in moderate latency (20 ms and 55 ms), whereas Client-Side Encryption recorded the highest latency (25 ms and 68 ms). This is indicative, that TDE provides the best performance among the encryption methods, while client-side encryption offers greater security at the expense of higher response times, particularly for OLAP workloads.

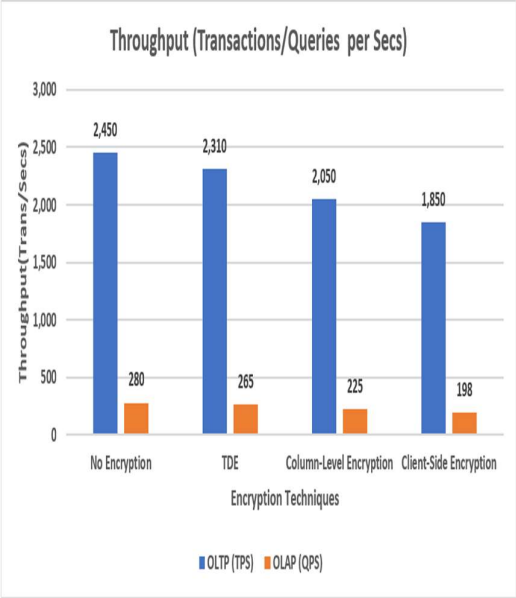

Figure 2: Throughput (Transactions/Queries per Second)

The experimental result of Throughput (Transactions/Queries per Second), which is shown in Figure 2, is indicative that database throughput decreases as encryption complexity increases. No Encryption achieved the highest throughput, recording 2,450 TPS for OLTP and 280 QPS for OLAP. TDE maintained relatively high performance with 2,310 TPS and 265 QPS, indicating minimal overhead. Column-Level Encryption reduced throughput to 2,050 TPS and 225 QPS, while Client-Side Encryption recorded the lowest throughput (1,850 TPS and 198 QPS) due to additional encryption and decryption operations. The result is indicative that TDE provides the best balance between security and throughput, whereas client-side encryption has the greatest impact on system performance.

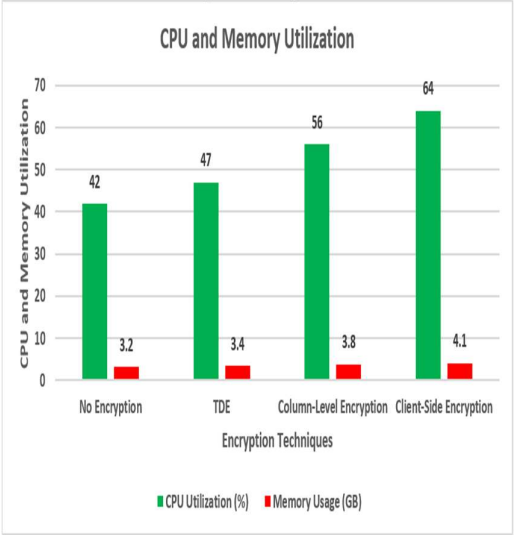

Figure 3: CPU and Memory Utilization

The experimental result of CPU and memory usage, based on of the three encryption techniques, as shown in Figure 3 indicates that CPU and memory usage increase with the complexity of the encryption technique. No Encryption recorded the lowest resource consumption, using 42% CPU and 3.2 GB of memory. TDE showed only a slight increase (47% CPU and 3.4 GB), while Column-Level Encryption required 56% CPU and 3.8 GB of memory. Client-Side Encryption consumed the most resources, with 64% CPU and 4.1 GB of memory. The results show that stronger encryption imposes greater computational overhead, with TDE remaining the most resource-efficient encryption method.

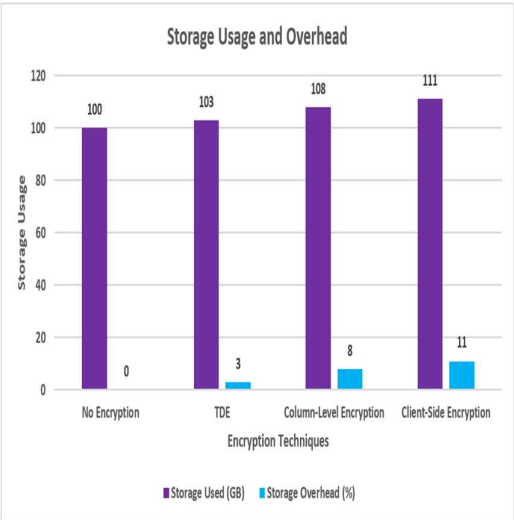

Figure 4: Storage Overhead

The result of experimentation for storage overhead based on the three encryption techniques applied, as shown in Figure 4 is indicative that storage requirements increase with the use of database encryption. No Encryption required 100 GB of storage with 0% overhead. TDE introduced a

minimal overhead, increasing storage to 103 GB (3%). Column-Level Encryption required 108 GB (8%), while Client-Side Encryption had the highest storage requirement at 111 GB (11%). The result show that TDE has the least impact on storage capacity, whereas client-side encryption incurs the greatest storage overhead due to additional encrypted data and metadata.

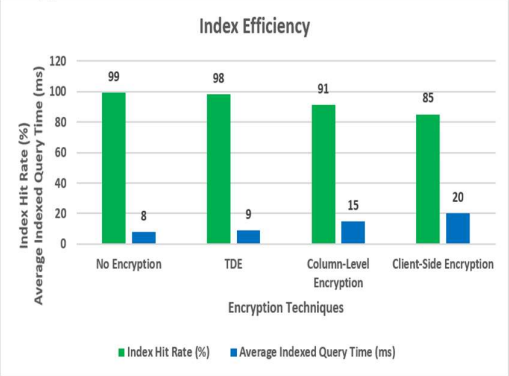

Figure 5: Index Efficiency

Index efficiency as an experimental output, as shown in Figure 5, illustrations that index efficiency decreases as encryption complexity increases. No Encryption achieved the highest index hit rate (99%) and the fastest indexed query time (8 ms). TDE had only a slight impact, with a 98% hit rate and 9 ms query time. Column-Level Encryption reduced the index hit rate to 91% and increased query time to 15 ms, while Client-Side Encryption recorded the lowest index efficiency, with an 85% hit rate and 20 ms query time.

This result is indicative that TDE preserves indexing performance more effectively than the other encryption techniques, whereas client-side encryption has the greatest impact on index efficiency.

Table 1: Performance Comparison across all encryption techniques

Metric	No Encryption	TDE	Column-Level Encryption	Client-Side Encryption
Query Latency (ms)	12	14	20	25
Throughput (TPS)	2,450	2,310	2,050	1,850
CPU Utilization (%)	42	47	56	64
Memory Usage (GB)	3.2	3.4	3.8	4.1
Storage Overhead (%)	0	3	8	11
Index Hit Rate (%)	99	98	91	85

Performance Comparison across all encryption techniques

Performance comparison across all encryption techniques applied, as shown in Table 1, provides an a complete comparison of the performance impact of the different database encryption techniques. No Encryption achieved the best performance across all metrics, recording the lowest query latency, highest throughput, lowest resource consumption, no storage overhead, and the highest index hit rate. Among the encryption methods, Transparent Data Encryption (TDE) showed the smallest performance impact, with only slight increases in latency, CPU and memory usage, and storage overhead while maintaining high throughput and index efficiency. Column-Level Encryption introduced moderate performance degradation, whereas Client-Side Encryption had the greatest impact, exhibiting the highest latency, resource utilization, and storage overhead, along with the lowest throughput and index efficiency. Summarily, the results indicate that TDE offers the best balance between data security and system performance, while client-side encryption provides stronger protection at the expense of greater performance overhead.

5. Discussion

The findings of this study demonstrate that database encryption significantly improves data security but introduces varying levels of performance overhead depending on the encryption technique and workload type. The benchmark evaluation showed clear differences among Transparent Data Encryption (TDE), Column-level encryption, and Client-side encryption in terms of query latency, throughput, resource utilization, storage overhead, and index efficiency.

1. The results showed that Transparent Data Encryption (TDE) consistently produced the lowest performance overhead among the encryption techniques. As shown in Figures 1 and 2, TDE recorded only a slight increase in query latency and a small reduction in throughput compared with the unencrypted database. This suggests that TDE efficiently encrypts data at rest without significantly affecting database operations.
2. In contrast, Column-level encryption introduced moderate performance degradation across all evaluation metrics. The increase in query latency and reduction in throughput indicate that encrypting selected database fields requires additional processing during query execution, particularly when encrypted columns are frequently accessed or indexed.

3. The study further found that Client-side encryption had the greatest impact on database performance. It recorded the highest query latency, the lowest throughput, the highest CPU and memory utilization, and the largest storage overhead. Index efficiency also declined considerably under client-side encryption. These results indicate that encrypting data before it reaches the database places additional processing demands on both the client application and database system. While this approach provides the strongest level of confidentiality because data remains encrypted even from the database server, the associated performance penalty makes it less suitable for high-volume transaction environments.
4. Another important finding is that OLAP workloads experienced greater performance degradation than OLTP workloads across all encryption techniques. Analytical queries involving joins, aggregations, and large data scans required more encryption and decryption operations than transactional queries, resulting in higher query latency and lower throughput. This suggests that workload characteristics significantly influence the effectiveness of database encryption strategies and should be considered when selecting an appropriate security solution.
5. The comparative evaluation also demonstrates that no single encryption technique is optimal for all database environments. Instead, organizations must balance security requirements with acceptable performance levels. TDE provides the best overall balance between security and efficiency, making it suitable for enterprise databases with high transaction volumes. Column-level encryption is appropriate when only specific sensitive fields require protection, while Client-side encryption is most suitable for highly sensitive applications where maximum confidentiality is prioritized over performance.

These findings achieved the objectives of the study by providing a comprehensive comparison of common database encryption techniques using a standardized benchmark framework. The results offer practical evidence that database encryption should be selected according to organizational security requirements, workload characteristics, and acceptable performance overhead. This contributes to existing knowledge by providing a unified evaluation of multiple encryption techniques across both OLTP and OLAP workloads, thereby supporting more informed database security decisions in modern enterprise and cloud environments.

6. Conclusion

This study evaluated the performance trade-offs of three widely used database encryption techniques, including Transparent Data Encryption (TDE), column-level encryption, and client-side encryption; using a benchmark framework based on OLTP and OLAP workloads. The evaluation considered key performance metrics, including query latency, throughput, CPU and memory utilization, storage overhead, and index efficiency.

The findings showed that while all encryption techniques enhance data security, they impose different levels of performance overhead. Transparent Data Encryption (TDE) consistently delivered the best performance, introducing only minimal increases in latency and resource usage while maintaining high throughput and index efficiency. Column-level encryption provided stronger protection for sensitive data with moderate performance degradation, whereas client-side encryption offered the highest level of confidentiality but incurred the greatest cost in terms of latency, resource consumption, storage overhead, and reduced throughput.

The study also found that analytical (OLAP) workloads were more affected by encryption than transactional (OLTP) workloads, highlighting the importance of considering workload characteristics when implementing database security solutions. These findings demonstrate that no single encryption technique is suitable for every application, and the choice of encryption should be guided by the required balance between security, performance, and operational needs.

The study provides a comprehensive comparison of common database encryption techniques and offers practical guidance for database administrators, system architects, and organizations in selecting appropriate encryption strategies. It also contributes to the existing literature by providing a unified benchmark-based evaluation of multiple encryption methods across different workload types.

7. Recommendations

Based on the findings, Transparent Data Encryption (TDE) is recommended for enterprise databases because it provides effective security with minimal performance overhead. Column-level encryption should be used when only sensitive data fields require protection, while client-side encryption is best suited for highly confidential applications where maximum security outweighs performance concerns.

Organizations should select encryption techniques based on their workload requirements, as OLAP workloads are more affected by encryption than OLTP workloads. Regular performance monitoring and the use of hardware-accelerated encryption technologies, such as AES-NI (Advanced

Encryption Standard New Instructions), are also recommended to reduce performance overhead.

8.Future Work

Future research should investigate hybrid encryption models that dynamically adjust encryption levels based on workload intensity and data sensitivity. Studies should also evaluate emerging technologies, including searchable encryption, homomorphic encryption, confidential computing, and AI-driven encryption optimization, to improve the balance between security and performance in cloud, hybrid, edge, and distributed database environments.

References

1) Ahsan, A. S., Arisa, E. Z., & Jilan, R. A. (2025). Evaluating database encryption: A comparative study of PostgreSQL implementations with EnterpriseDB TDE and Cybertec TDE. *Journal of Innovation and Technology Polbeng Series on Informatics, 10*(3), 1342–1349. https://doi.org/10.35314/1dp5mp08
2) Alenezi, M. N., Alabdulrazzaq, H., Alhatlani, H. M., Alobaid, F. A. S., et al. (2024). *On the performance of AES algorithm variants*. *International Journal of Information and Computer Security, 23*(3), 322–337. https://doi.org/10.1504/IJICS.2024.138494
3) Carvalho, M., Sá, F., & Bernardino, J. (2025). Evaluation of the impact of AES encryption on query read performance across Oracle, MySQL, and SQL Server databases. *Cryptography, 9*(4), 77. https://doi.org/10.3390/cryptography9040077
4) European Parliament and the Council of the European Union[EP-CEU].(2016). *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation).* Official Journal of the European Union.
5) Federal Republic of Nigeria[FRN].(2023). *Nigeria Data Protection Act, 2023 (Act No. 37 of 2023).* Federal Government Printer.
6) Khan, F. A., Jamjoom, M., Ahmad, A., & Asif, M. (2022). *An analytic study of architecture, security, privacy, query processing, and performance evaluation of database-as-a-service. Transactions on Emerging Telecommunications Technologies, 33*(2), e3814. https://doi.org/10.1002/ett.3814
7) Kim, E., & Jeon, S. (2025). Performance analysis of AES, RSA, and hybrid-based database encryption and decryption. *Journal of Convergence Security, 25*(4), 207–216. https://doi.org/10.33778/kcsa.2025.25.4.207
8) Lee, S., & Hur, J. (2019). Analysis of encryption algorithm performance by workload in big data platforms. *Journal of the Korea Institute of Information Security & Cryptology, 29*(6), 1305–1317. https://doi.org/10.13089/JKIISC.2019.29.6.1305
9) Madyatmadja, E. D., Hakim, A. N., & Sembiring, D. J. M. (2021**).** *Performance testing on Transparent Data Encryption for SQL Server's reliability and efficiency. Journal of Big Data, 8*, 134. https://doi.org/10.1186/s40537-021-00520-z
10) Mattsson, U. T. (2005). *Database encryption: How to balance security with performance. SSRN Electronic Journal.* https://doi.org/10.2139/ssrn.670561
11) Microsoft. (2025a). *Transparent Data Encryption (TDE).* Microsoft Learn.
12) Microsoft. (2025b). *Always Encrypted (Database Engine).* Microsoft Learn.
13) National Institute of Standards and Technology.[NIST] (2023). *Advanced Encryption Standard (AES) (FIPS PUB 197).* https://doi.org/10.6028/NIST.FIPS.197-upd1
14) National Institute of Standards and Technology[NIST].(2007). *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC* (Special Publication 800-38D). https://doi.org/10.6028/NIST.SP.800-38D
15) Oracle.(2025). *General considerations of using Transparent Data Encryption.* Oracle Database Documentation.
16) PCI Security Standards Council.[PCI-SSC] (2022). *Payment Card Industry Data Security Standard (PCI DSS), Version 4.0.*
17) Popa, R. A., Redfield, C. M. S., Zeldovich, N., & Balakrishnan, H. (2011). *CryptDB: Protecting confidentiality with encrypted query processing.* In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles (SOSP '11)* (pp. 85–100). Association for Computing Machinery. https://doi.org/10.1145/2043556.2043566 (DBLP)
18) PostgreSQL Global Development Group[PostgreSQLGDG]. (2025). *Encryption options.* PostgreSQL Documentation.
19) Sharma, P., Govorov, M., & Martin, M. (2024). A usable encryption solution for file-based geospatial data within a database file system. *Journal of Cybersecurity and Privacy, 4*(2), 298–323.
20) Sun, B., Zhao, S., & Tian, G. (2024). *SQL queries over encrypted databases: A survey*. *Connection Science*, 36(1).

https://doi.org/10.1080/09540091.2024.2323059

21) Tawose, O. T., Dai, J., Yang, L., & Zhao, D. (2023). *Toward efficient homomorphic encryption for outsourced databases through parallel caching. Proceedings of the ACM on Management of Data, 1*(2), Article 179. https://doi.org/10.1145/3589295
22) U.S. Department of Health and Human Services.[USDHHS] (2013). *Health Insurance Portability and Accountability Act (HIPAA) Security Rule.*
23) Vidaković, A., Petrović, T., Kresoja, P., & Veinović, M. (2024). Impact of database encryption on web application performance. In *Proceedings of the Sinteza 2024 International Scientific Conference on Information Technology and Data Related Research* (pp. 82–87). https://doi.org/10.15308/Sinteza-2024-82-87