IJEES

International Journal of Electrical Engineering and Sustainability (IJEES)

ISSN (online): 2959-9229

https://ijees.org/index.php/ijees/index

ISI 2023-2024: (0.557) Arab Impact Factor: 1.51 SJIF 2024 = 5.274

Volume 3 | Number 1 | January-March 2025 | Pages 28-39

Research Article

Future Trends and Real-World Applications in Database Encryption

Emhemed Mohamed [1*]

[1] Department of Computer Science, Faculty of Science, Bani Walid University, Bani Walid, Libya

*Corresponding author: emhemed82@gmail.com

Received: October 30, 2024 Accepted: December 23, 2024 Published: January 13, 2025

Abstract: As data becomes the cornerstone of digital ecosystems, securing databases against evolving cyber threats is imperative. This paper explores the critical role of encryption in safeguarding sensitive information, achieving regulatory compliance, and maintaining data integrity. It provides a comprehensive analysis of foundational encryption techniques, including symmetric, asymmetric, and advanced methods like homomorphic and format-preserving encryption. Real-world applications across industries such as finance, healthcare, and government highlight encryption's effectiveness in addressing specific security needs. The paper delves into emerging trends, such as quantum-resistant cryptography and privacy-preserving computations, emphasizing their potential to redefine database security. While encryption offers robust protection, challenges like performance overhead, key management complexities, and query limitations demand innovative solutions. Strategies such as efficient algorithms, hardware acceleration, and robust Key Management Systems (KMS) are discussed as means to balance security and performance. This paper underscores the necessity of adopting encryption best practices as part of a holistic security framework, urging organizations to proactively integrate these advancements to safeguard their databases in an increasingly data-driven and interconnected world.

Keywords: Database Encryption, Security and Performance Balance, Key Management Systems (KMS), Quantum-Resistant Cryptography, Data Protection Best Practices.

1. Introduction

In today's interconnected and data-driven world, databases are the backbone of modern organizations, storing critical information such as customer records, financial data, intellectual property, and operational insights. However, the value of this information makes it a prime target for cyberattacks, ranging from insider threats and ransomware to sophisticated hacking attempts by nation-states [1]. As the volume of data grows and digital ecosystems expand, so too does the urgency to secure these repositories against unauthorized access and exploitation.

Database security encompasses a wide range of measures designed to protect stored data from threats, including physical breaches, software vulnerabilities, and human error. Among these measures, encryption stands out as a vital tool for ensuring confidentiality and protecting sensitive information. By converting plaintext data into unreadable ciphertext using cryptographic algorithms, encryption ensures that even if attackers gain access to the database, the information remains unintelligible without the appropriate decryption keys [2]. This makes encryption not only a preventive measure but also a fail-safe mechanism in scenarios where perimeter defenses are compromised.

The role of encryption in database security extends beyond protecting against external threats. It is also crucial for meeting regulatory and compliance requirements, such as the General Data Protection Regulation (GDPR), the Health Insurance Portability and Accountability Act (HIPAA), and the Payment Card Industry Data Security Standard (PCI DSS). These frameworks often mandate encryption for sensitive data to protect individual privacy and secure financial transactions. Beyond compliance, encryption fosters trust by demonstrating an organization's commitment to safeguarding its stakeholders' data [4,5].

Despite its importance, encryption in database security is not without its challenges. Implementing robust encryption requires careful planning to balance security, performance, and operational efficiency. For instance, encrypted databases may face slower query performance, complications with data indexing, and increased complexity in managing encryption keys. Additionally, emerging technologies and advanced attack methods continually test the robustness of existing encryption strategies, demanding continuous innovation and adaptation. This article seeks to provide a comprehensive overview of database encryption by discussing its fundamental concepts and technical intricacies. It will explore the different encryption techniques, their application in securing databases, and the technical hurdles organizations encounter during implementation. By examining these aspects, the article aims to equip readers with the knowledge needed to understand the trade-offs, limitations, and potential of encryption as a pivotal component of database security.

2. Fundamentals of Database Encryption

A. Definition of Encryption in the Context of Databases

Encryption in the realm of database security is the process of converting plaintext data into ciphertext using cryptographic algorithms, rendering the information indecipherable without the appropriate decryption key. This transformative process ensures that sensitive data remains protected, even when accessed by unauthorized entities. Within databases, encryption operates at various levels—ranging from individual fields to entire datasets—integrating cryptographic safeguards to mitigate risks associated with data breaches, unauthorized access, and cyberattacks [8,9].

In essence, database encryption serves as a critical layer of defense, ensuring that data confidentiality is preserved irrespective of whether it resides in storage, transit, or active use. By doing so, encryption not only protects the integrity of sensitive information but also supports organizations in adhering to stringent data privacy regulations [10-15].

B. Key Purposes of Database Encryption

- Protecting Sensitive Information

In an age where data serves as an invaluable asset, safeguarding its confidentiality is paramount. Databases often store personally identifiable information (PII), financial records, intellectual property, and other sensitive datasets. Encryption ensures that such data remains inaccessible to unauthorized users, even in the event of a breach or compromise. By obscuring the content of the database, encryption minimizes the potential impact of cyberattacks, such as ransomware and data exfiltration.

- Compliance with Data Privacy Regulations

Regulatory frameworks such as the General Data Protection Regulation (GDPR), the Health Insurance Portability and Accountability Act (HIPAA), and the Payment Card Industry Data Security Standard (PCI DSS) mandate robust data protection measures, including encryption. These regulations are designed to uphold individual privacy, protect sensitive data, and promote accountability among organizations handling critical information. By implementing encryption, organizations can demonstrate compliance with these standards, thereby avoiding penalties and reinforcing stakeholder trust.

C. Types of Data Subject to Encryption

- Data at Rest

Data at rest refers to information stored in databases, files, or backup systems that is not actively being accessed or transmitted. Examples include databases hosted on physical servers, virtual machines, or cloud storage solutions. Encrypting data at rest protects it from unauthorized access resulting from physical theft, unauthorized database queries, or misconfigurations in storage environments. Techniques such as Transparent Data Encryption (TDE) and full-disk encryption are commonly employed to secure data at rest.

- Data in Transit

Data in transit pertains to information being transmitted between systems, applications, or networks. This includes data exchanged between database servers and client applications or between distributed nodes in cloud-based systems. Encrypting data in transit prevents interception and eavesdropping attacks, such as man-in-the-middle (MITM) attacks. Secure protocols like Transport Layer Security (TLS) and IPsec are often utilized to ensure that transmitted data remains confidential and tamper-proof.

- Data in Use

Data in use refers to information actively being processed, analyzed, or updated within applications or systems. Encrypting data in use presents unique challenges, as it requires maintaining confidentiality without disrupting operational workflows. Emerging techniques like homomorphic encryption and secure enclaves allow computations to be performed on encrypted data without the need for decryption, preserving security while enabling analytics and real-time processing.

By addressing all three states of data—at rest, in transit, and in use—database encryption establishes a comprehensive security framework that safeguards sensitive information throughout its lifecycle. This multi-dimensional approach is critical for ensuring data confidentiality, integrity, and compliance in a rapidly evolving digital ecosystem.

3. Core Concepts of Database Encryption

Database encryption is a cornerstone of modern data security, providing robust protection for sensitive information against unauthorized access and cyber threats. It involves converting data into an unreadable format using cryptographic techniques, ensuring that only authorized entities with decryption keys can access the original information. This introduction explores the fundamental concepts of database encryption through its key components: encryption techniques, encryption levels, Transparent Data Encryption (TDE), end-to-end encryption, and advanced encryption methodologies. These concepts collectively enable organizations to implement comprehensive security strategies tailored to their specific data protection requirements [16-23]. Figure 1 illustrates these core concepts of database encryption, highlighting their interconnections and practical applications.

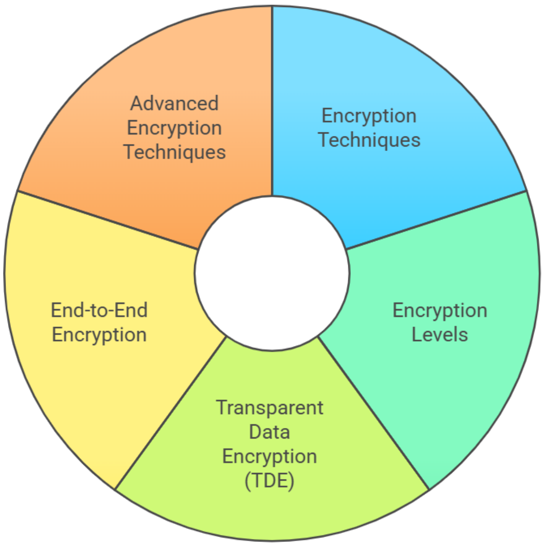
Advanced Encryption Techniques
Encryption Techniques
End-to-End Encryption
Encryption Levels
Transparent Data Encryption (TDE)

Figure 1. Core concepts of database encryption.

A. Encryption Techniques

- Symmetric Encryption

Symmetric encryption relies on a single key for both encryption and decryption processes. This technique is widely used in database encryption due to its efficiency in handling large volumes of data. The Advanced Encryption Standard (AES) is a prominent symmetric encryption algorithm known for its robust security and performance. AES offers various key lengths (e.g., 128, 192, and 256 bits) to balance security and computational overhead. While symmetric encryption is highly efficient, key management becomes critical since the shared key must be securely distributed and stored.

- Asymmetric Encryption

Asymmetric encryption uses a pair of keys: a public key for encryption and a private key for decryption. This technique is often employed in scenarios requiring secure communication and key exchange. RSA (Rivest-Shamir-Adleman) is a widely used asymmetric encryption algorithm. While more computationally intensive than symmetric encryption, it eliminates the need for shared secret keys. Asymmetric encryption is typically used in hybrid encryption systems, where it secures key exchanges while symmetric encryption handles bulk data encryption.

- Hashing for Data Integrity

Hashing is a cryptographic technique that transforms data into a fixed-length hash value, ensuring data integrity and authenticity. Unlike encryption, hashing is a one-way process, meaning the original data cannot be retrieved from the hash. Secure Hash Algorithms (e.g., SHA-256) are commonly used to verify data integrity by comparing stored and computed hash values. While hashing does not directly encrypt data, it plays a crucial role in detecting unauthorized modifications.

B. *Encryption Levels*

- Full-Database Encryption

Full-database encryption secures the entire database, including all its tables, indexes, and logs. This approach simplifies implementation but may impact query performance due to the broad scope of encryption. Full-database encryption is suitable for environments where all data must be protected, such as compliance with stringent regulatory requirements.

- Table-Level Encryption

Table-level encryption targets specific tables within a database. It provides more granularity than full-database encryption, allowing organizations to prioritize sensitive data. For example, a database containing both public and confidential records can encrypt only the tables holding sensitive information, optimizing performance without compromising security.

- Column-Level Encryption

Column-level encryption focuses on encrypting specific fields or columns within a table, such as personally identifiable information (PII) or credit card numbers. This method offers the highest level of granularity, enabling organizations to secure sensitive data while leaving less critical information unencrypted. However, managing column-level encryption can introduce complexity, particularly in terms of key management and query optimization.

C. *Transparent Data Encryption (TDE)*

- Features and Benefits of TDE

Transparent Data Encryption (TDE) is a database-level encryption method that automatically encrypts data at rest without requiring changes to applications. TDE encrypts the database files and backups, ensuring protection against unauthorized access or physical theft. Key benefits of TDE include:

- Ease of Implementation: Requires minimal changes to existing database configurations.
- Performance Optimization: Designed to minimize overhead during encryption and decryption processes.
- Compliance Readiness: Aligns with regulations mandating encryption of data at rest.
- While TDE offers robust protection for stored data, it does not encrypt data in transit or in use, requiring complementary security measures for comprehensive protection.

D. *End-to-End Encryption*

End-to-end encryption ensures that data remains encrypted from the moment it is created until it reaches its final destination. This comprehensive approach prevents data exposure during transmission or storage. End-to-end encryption is particularly vital in distributed environments, such as cloud-based databases and multi-tenant architectures. By encrypting data at every stage, organizations can safeguard information against eavesdropping, interception, and unauthorized access. However, implementing end-to-end encryption requires careful integration to balance usability and security.

E. Advanced Encryption Techniques

- Homomorphic Encryption

Homomorphic encryption is an advanced technique that allows computations to be performed on encrypted data without decrypting it. This enables secure data processing in untrusted environments, such as third-party cloud platforms, while preserving confidentiality. Although homomorphic encryption is computationally intensive, ongoing advancements aim to make it more practical for real-world applications.

- Format-Preserving Encryption

Format-preserving encryption (FPE) encrypts data while maintaining its original format. For instance, an encrypted credit card number retains the same structure (e.g., a 16-digit sequence), enabling compatibility with legacy systems and existing workflows. FPE is particularly useful in scenarios where encrypted data must adhere to specific formatting requirements, such as regulatory or industry standards. By leveraging these core concepts and techniques, database encryption provides a robust framework for securing sensitive information across its lifecycle, aligning with both operational and regulatory demands.

4. Technical Challenges in Database Encryption

Database encryption is a critical component of modern cybersecurity frameworks, ensuring the confidentiality and integrity of sensitive information. However, while encryption offers robust protection against unauthorized access, its implementation is not without challenges. These challenges arise from the intricate balance between security, performance, and operational efficiency, particularly in environments with diverse technological infrastructures and regulatory demands. This introduction examines the key technical challenges associated with database encryption, categorized into performance overhead, key management, compatibility with legacy systems, query limitations, compliance and regulatory requirements, backup and recovery, security risks, and scalability in multi-tenant environments [24-30]. Figure 2 demonstrates the technical challenges in database encryption.

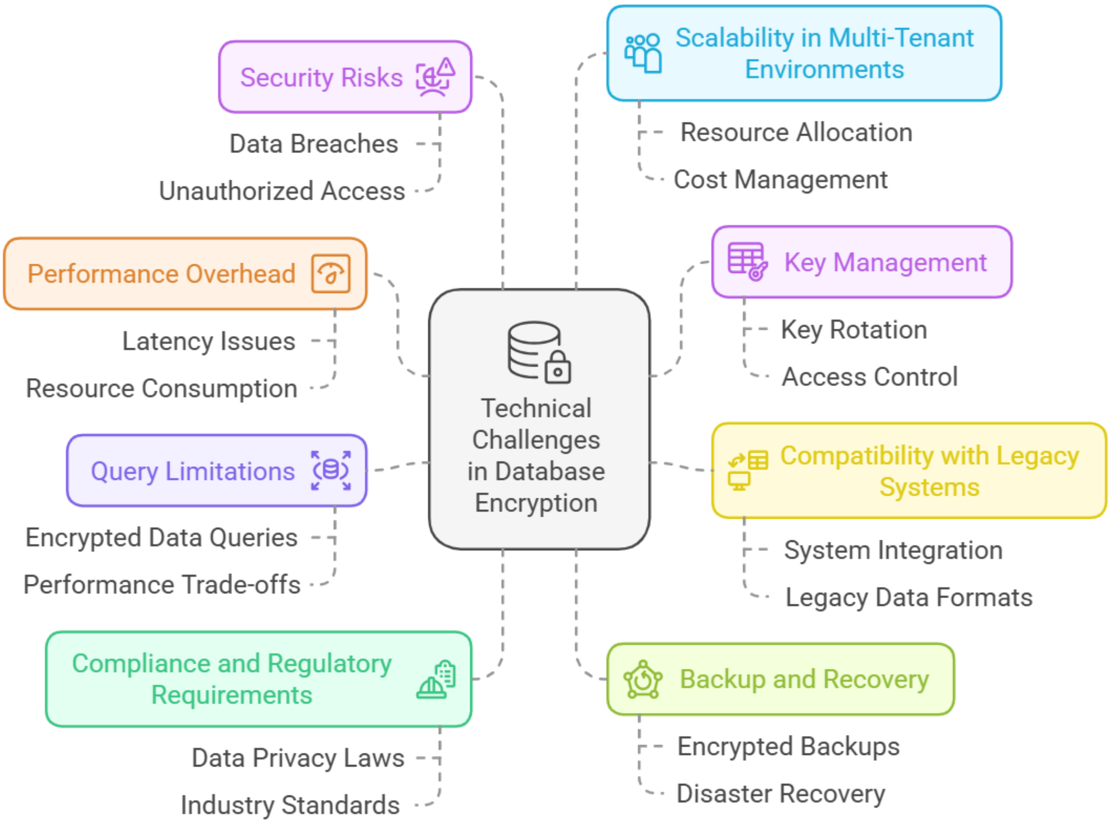

Figure 2. The technical challenges in database encryption.

A. *Performance Overhead*

One of the most significant challenges in implementing database encryption is the impact on performance. Encrypting data adds computational overhead during query execution and transaction processing. This is particularly evident in environments where real-time data access is critical, such as financial systems or e-commerce platforms. The process of encrypting and decrypting data during read/write operations can slow response times, potentially affecting user experience. Moreover, the encryption of indexes can hinder the efficiency of database queries, resulting in slower searches and increased resource consumption.

B. *Key Management*

Key management is a cornerstone of encryption security, yet it presents a complex challenge. Organizations must ensure the secure storage, rotation, and revocation of encryption keys. Loss of keys can render data irretrievable, while poorly secured keys increase the risk of unauthorized access. In distributed systems and multi-cloud environments, managing keys across multiple locations further complicates the process. Automated key management systems (KMS) can mitigate some of these issues, but their deployment and maintenance require substantial investment and expertise.

C. *Compatibility with Legacy Systems*

Integrating encryption with legacy databases and applications poses significant hurdles. Many older systems were not designed with encryption in mind and may lack support for modern cryptographic techniques. Retrofitting these systems to accommodate encryption can result in operational disruptions, increased costs, and compatibility issues. In some cases, upgrading or replacing legacy systems becomes necessary, adding complexity to the implementation process.

D. *Query Limitations*

Encryption often restricts the use of advanced database features, such as indexing, searching, and sorting. Encrypted data is typically stored as ciphertext, which cannot be processed by conventional query operations. This limitation requires additional workarounds, such as partial decryption or specialized cryptographic techniques, which can degrade performance and complicate application logic. For example, full-text search on encrypted fields may necessitate costly preprocessing steps, reducing overall efficiency.

E. *Compliance and Regulatory Requirements*

While encryption is a critical component of regulatory compliance, navigating the diverse requirements of frameworks like GDPR, HIPAA, and PCI DSS can be challenging. Each regulation imposes specific standards for encryption algorithms, key management, and audit trails. Ensuring compliance across multiple jurisdictions and industries demands continuous monitoring and adaptation of encryption practices. Non-compliance can result in severe penalties, reputational damage, and legal liabilities.

F. *Backup and Recovery*

Encrypting database backups is essential to protect data against unauthorized access during storage or transit. However, encrypted backups introduce complications in disaster recovery scenarios. Organizations must ensure that decryption keys are securely stored and readily accessible during recovery processes. Additionally, managing the synchronization of encryption keys with backup systems can be complex, especially in environments with frequent data updates.

G. *Security Risks*

Despite its advantages, encryption is not immune to security risks. Attackers often target encryption keys through techniques like phishing, malware, or side-channel attacks. Side-channel attacks, in particular, exploit information leaks from cryptographic implementations, such as power consumption or timing variations, to infer keys. Moreover, misconfigurations, weak algorithms, or improper key management can undermine the effectiveness of encryption, leaving databases vulnerable.

H. *Scalability in Multi-Tenant Environments*

In cloud-hosted or multi-tenant database environments, achieving scalable encryption presents unique challenges. Encrypting data for multiple tenants requires isolating encryption keys to prevent unauthorized access between tenants. Additionally, performance overhead becomes more pronounced as the number of tenants and encrypted datasets increases. Cloud service providers must implement robust encryption practices while maintaining high availability and efficiency, a balance that is often difficult to achieve.

Addressing these technical challenges requires a combination of robust encryption algorithms, effective key management practices, and thoughtful system design. By understanding and mitigating these issues, organizations can enhance their database security while minimizing the trade-offs associated with encryption.

5. Strategies to Address Challenges in Database Encryption

A. Using Efficient Encryption Algorithms and Hardware Acceleration

The choice of encryption algorithms significantly impacts performance and security. Efficient algorithms, such as the Advanced Encryption Standard (AES), provide strong protection while minimizing computational overhead. AES, with its hardware support (e.g., AES-NI on modern processors), allows encryption and decryption to be performed faster, reducing latency in database operations [31-35].

Hardware acceleration further enhances performance by offloading cryptographic computations to specialized hardware components, such as Trusted Platform Modules (TPMs) or Hardware Security Modules (HSMs). These devices not only speed up encryption processes but also provide secure environments for key storage and cryptographic operations.

B. Implementing Robust Key Management Systems (KMS)

Key management is central to the effectiveness of encryption. Implementing a robust Key Management System (KMS) ensures secure storage, generation, rotation, and revocation of encryption keys. Best practices for key management include:

- Centralized Key Management: Using a unified system to manage keys across databases and environments.
- Key Rotation Policies: Periodically updating keys to reduce the risk of compromise.
- Access Control: Enforcing strict access policies to prevent unauthorized users from accessing encryption keys.
- Backup and Recovery: Securely backing up keys to ensure availability during disaster recovery scenarios.

Many cloud providers, such as AWS KMS and Azure Key Vault, offer integrated KMS solutions that simplify the process for organizations operating in distributed or multi-cloud environments.

C. Optimizing Database Performance for Encrypted Data

To mitigate the performance overhead associated with encryption, organizations can optimize database configurations and workflows:

- Partitioning and Indexing: Ensuring that encrypted databases are partitioned and indexed effectively to improve query performance.
- Caching: Leveraging caching mechanisms to reduce the frequency of encryption and decryption operations for frequently accessed data.
- Query Optimization: Writing efficient queries that minimize the need for complex operations on encrypted fields.
- Selective Encryption: Encrypting only sensitive data instead of encrypting entire datasets, balancing security with performance.

D. Leveraging Hybrid Encryption Techniques

Hybrid encryption combines the strengths of symmetric and asymmetric encryption to achieve a balance between security and efficiency. For example:

- Symmetric Encryption: Used for encrypting large datasets due to its speed and low computational overhead.
- Asymmetric Encryption: Used for securely transmitting symmetric keys.

This approach is particularly useful in distributed systems where secure key exchange is critical. By leveraging hybrid encryption, organizations can ensure robust data protection without compromising performance.

E. *Regularly Updating and Auditing Encryption Mechanisms*

Encryption technologies and algorithms must evolve to address emerging threats and vulnerabilities. Regular updates and audits are essential to maintain the security and effectiveness of encryption implementations:

- Algorithm Updates: Replacing deprecated or vulnerable algorithms with modern, secure alternatives (e.g., migrating from SHA-1 to SHA-256).
- Security Audits: Conducting periodic reviews to identify potential vulnerabilities in encryption configurations and practices.
- Compliance Checks: Ensuring that encryption mechanisms meet the latest regulatory and industry standards.
- Threat Monitoring: Staying informed about new attack methods, such as quantum computing threats, and preparing to adopt quantum-resistant cryptographic solutions.

By adopting these strategies, organizations can address the technical challenges of database encryption, ensuring robust protection for sensitive data while maintaining operational efficiency. Combining advanced encryption techniques with sound management practices positions databases to withstand evolving security threats in a rapidly changing digital landscape.

6. Real-World Applications

A. *Examples of Organizations or Industries Using Database Encryption*

- Financial Services:

Banks and financial institutions routinely use database encryption to protect sensitive customer data, such as account information, credit card details, and transaction records. For example, JPMorgan Chase employs advanced encryption protocols to ensure compliance with regulations like PCI DSS and safeguard against cyberattacks.

- Healthcare:

Healthcare providers and organizations secure electronic health records (EHRs) using encryption to comply with HIPAA regulations. For instance, the Mayo Clinic integrates database encryption with role-based access controls to protect patient information from unauthorized access.

- E-Commerce and Retail:

E-commerce platforms like Amazon use encryption to protect customer payment details, ensuring secure transactions. Database encryption also helps meet GDPR requirements for handling European customer data.

- Government Agencies:

Governments worldwide use encryption to protect classified information and citizen data. For example, the U.S. Department of Defense employs military-grade encryption to safeguard sensitive intelligence.

- Cloud Service Providers:

Cloud platforms like AWS, Microsoft Azure, and Google Cloud offer built-in encryption services for databases, such as Transparent Data Encryption (TDE) and customer-managed keys, enabling businesses to secure data hosted in the cloud.

B. *Lessons Learned from Successful Implementations*

- Layered Security is Essential:

Encryption is most effective when combined with other security measures, such as firewalls, intrusion detection systems, and role-based access controls.

- Key Management is Critical:

Organizations that adopt robust Key Management Systems (KMS) experience fewer incidents of key compromise or loss, ensuring sustained data availability and security.

- Scalability Matters:

Companies with scalable encryption solutions can adapt to growing data volumes and evolving compliance requirements without compromising performance.

- Compliance Drives Adoption:

Successful implementations align encryption practices with regulatory requirements, avoiding penalties and fostering customer trust.

- Proactive Monitoring and Auditing are Crucial:

Continuous monitoring and periodic security audits help organizations identify vulnerabilities and update their encryption protocols as needed.

7. Future Trends in Database Encryption

A. Advances in Cryptographic Algorithms

The development of faster, more secure cryptographic algorithms continues to shape the future of database encryption. For example:

- Post-Quantum Cryptography: As quantum computing becomes a reality, traditional encryption algorithms like RSA and ECC face potential vulnerabilities. Post-quantum algorithms, such as lattice-based and hash-based cryptography, are being developed to withstand quantum attacks.
- Lightweight Cryptography: Designed for resource-constrained environments, such as IoT devices, lightweight encryption algorithms optimize security without significant performance trade-offs.

B. Emerging Technologies Like Quantum-Resistant Encryption

Quantum computing poses a significant threat to current cryptographic standards, necessitating the development of quantum-resistant encryption techniques. Organizations are beginning to experiment with algorithms that leverage complex mathematical problems resistant to quantum attacks, such as:

- Lattice-Based Cryptography: Ensures security by constructing cryptographic systems based on lattice problems, which are computationally infeasible to solve with quantum algorithms.
- Multivariate Cryptography: Employs multivariate polynomial equations as a foundation for quantum-resistant encryption.
- Governments and industries are already incorporating quantum-resistant algorithms into their research and planning to future-proof encryption systems.

C. Increased Adoption of Homomorphic Encryption for Data Analytics

Homomorphic encryption is gaining traction for its ability to perform computations on encrypted data without decryption. This innovation allows organizations to analyze sensitive data securely while maintaining confidentiality, particularly in:

- Healthcare: Enabling secure analysis of patient data across institutions without compromising privacy.
- Financial Services: Allowing encrypted financial modeling and fraud detection in cloud-based environments.
- AI and Machine Learning: Supporting privacy-preserving training and inference on sensitive datasets.
- Although homomorphic encryption is currently computationally intensive, ongoing advancements aim to make it more efficient and practical for widespread adoption.

The future of database encryption lies in integrating these advancements with existing systems to address emerging security challenges. By staying ahead of technological trends and adopting innovative encryption techniques, organizations can ensure the long-term security and resilience of their databases.

8. Conclusion

Database encryption has become a cornerstone of modern security strategies, providing essential protection for sensitive information against the ever-evolving landscape of cyber threats. This article explored the fundamental concepts of database encryption, its key purposes in protecting sensitive data and achieving regulatory compliance, and the technical challenges it poses, such as performance overhead, key management, and query limitations.

The discussion highlighted core encryption techniques, including symmetric and asymmetric encryption, hashing for data integrity, and advanced methods like homomorphic and format-preserving encryption. It also underscored the practical implementation levels, such as full-database, table-level, and column-level encryption, along with the significance of tools like Transparent Data Encryption (TDE) and end-to-end encryption. Real-world applications and lessons from industries such as finance, healthcare, and government demonstrated the effectiveness of encryption in addressing specific use cases. Moreover, emerging trends, including quantum-resistant cryptography and homomorphic encryption, underscore the future direction of database security.

Despite its benefits, encryption requires a careful balance between security and performance. Excessive encryption can degrade system efficiency, while insufficient measures leave databases vulnerable. Addressing these challenges necessitates adopting efficient algorithms, leveraging hardware acceleration, optimizing database performance, and implementing robust Key Management Systems (KMS). Regular updates, security audits, and compliance monitoring are essential for maintaining a secure and resilient database ecosystem.

As data continues to grow in volume and value, the importance of encryption cannot be overstated. Organizations must proactively adopt encryption best practices to protect their assets, comply with regulations, and foster trust with stakeholders. By integrating encryption into a broader security strategy and staying abreast of technological advancements, businesses can ensure the confidentiality, integrity, and availability of their databases in an increasingly interconnected and data-driven world.

Author Contributions: Author has contributed significantly to the development and completion of this article.

Funding: This article received no external funding.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to express their sincere gratitude to Department of Computer Science, Faculty of Science, Bani Walid University, Bani Walid, Libya, for their invaluable support and resources throughout the course of this research.

Conflicts of Interest: The author(s) declare no conflict of interest.

ORCID

Emhemed Mohamed https://orcid.org/0009-0009-8676-5381

References

[1] M. Das, X. Tao, and J. C. P. Cheng, "BIM security: A critical review and recommendations using encryption strategy and blockchain," *Autom. Constr.*, vol. 126, no. 103682, p. 103682, 2021.

[2] S. Yadala, C. S. R. Pundru, and V. K. Solanki, "A novel private encryption model in IoT under cloud computing domain," in *Lecture Notes in Networks and Systems*, Singapore: Springer Nature Singapore, 2023, pp. 263–270.

[3] M. Khaleel, A. Jebrel, and D. M. Shwehdy, "*Artificial intelligence in computer science,*" Int. J. Electr. Eng. and Sustain., pp. 01–21, 2024.

[4] S. Yin, H. Li, L. Teng, A. A. Laghari, and V. V. Estrela, "Attribute-based multiparty searchable encryption model for privacy protection of text data," *Multimed. Tools Appl.*, vol. 83, no. 15, pp. 45881–45902, 2023.

[5] M. SaberiKamarposhti, A. Ghorbani, and M. Yadollahi, "A comprehensive survey on image encryption: Taxonomy, challenges, and future directions," *Chaos Solitons Fractals*, vol. 178, no. 114361, p. 114361, 2024.

[6] S. Ahmad and S. Mehfuz, "Efficient time-oriented latency-based secure data encryption for cloud storage," *Cyber Security and Applications*, vol. 2, no. 100027, p. 100027, 2024.

[7] E. Gokcay and H. Tora, "A novel data encryption method using an interlaced chaotic transform," *Expert Syst. Appl.*, vol. 237, no. 121494, p. 121494, 2024.

[8] A. Ali, B. A. S. Al-rimy, F. S. Alsubaei, A. A. Almazroi, and A. A. Almazroi, "HealthLock: Blockchain-based privacy preservation using homomorphic encryption in Internet of Things healthcare applications," *Sensors (Basel)*, vol. 23, no. 15, p. 6762, 2023.

[9] R. Walid, K. P. Joshi, and S. G. Choi, "Comparison of attribute-based encryption schemes in securing healthcare systems," *Sci. Rep.*, vol. 14, no. 1, pp. 1–14, 2024.

[10] B. Seth, S. Dalal, V. Jaglan, D.-N. Le, S. Mohan, and G. Srivastava, "Integrating encryption techniques for secure data storage in the cloud," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 4, 2022.

[11] C. Wang, D. Tang, H. Lin, F. Yu, and Y. Sun, "High-dimensional memristive neural network and its application in commercial data encryption communication," *Expert Syst. Appl.*, vol. 242, no. 122513, p. 122513, 2024.

[12] S. Das and S. Namasudra, "A novel hybrid encryption method to secure healthcare data in IoT-enabled healthcare infrastructure," *Comput. Electr. Eng.*, vol. 101, no. 107991, p. 107991, 2022.

[13] K. O.-B. O. Agyekum, Q. Xia, E. B. Sifah, C. N. A. Cobblah, H. Xia, and J. Gao, "A proxy re-encryption approach to secure data sharing in the internet of things based on blockchain," *IEEE Syst. J.*, vol. 16, no. 1, pp. 1685–1696, 2022.

[14] M. Khaleel, A. Ahmed, and A. H. Alsharif, "Artificial Intelligence in Engineering," *Brill Res Artif Intell*, vol. 3, no. 1, pp. 32–42, 2023.

[15] R. Denis and P. Madhubala, "Hybrid data encryption model integrating multi-objective adaptive genetic algorithm for secure medical data communication over cloud-based healthcare systems," *Multimed. Tools Appl.*, vol. 80, no. 14, pp. 21165–21202, 2021.

[16] R. Roselinkiruba and G. Bhuvaneshwari, "Feature extraction based pixel segmentation techniques data hiding and data encryption," *Multimed. Tools Appl.*, vol. 83, no. 7, pp. 19259–19276, 2023.

[17] S. A. Rieyan *et al.*, "An advanced data fabric architecture leveraging homomorphic encryption and federated learning," *Inf. Fusion*, vol. 102, no. 102004, p. 102004, 2024.

[18] V. Terziyan, B. Bilokon, and M. Gavriushenko, "Deep homeomorphic data encryption for privacy preserving machine learning," *Procedia Comput. Sci.*, vol. 232, pp. 2201–2212, 2024.

[19] M. K. Hasan *et al.*, "Lightweight encryption technique to enhance medical image security on internet of medical things applications," *IEEE Access*, vol. 9, pp. 47731–47742, 2021.

[20] A. Sasikumar *et al.*, "Blockchain-assisted hierarchical attribute-based encryption scheme for secure information sharing in industrial internet of things," *IEEE Access*, vol. 12, pp. 12586–12601, 2024.

[21] M. N. Ramachandra, M. Srinivasa Rao, W. C. Lai, B. D. Parameshachari, J. Ananda Babu, and K. L. Hemalatha, "An efficient and secure big data storage in Cloud environment by using Triple Data Encryption Standard," *Big Data Cogn. Comput.*, vol. 6, no. 4, p. 101, 2022.

[22] B. Li, Y. Feng, Z. Xiong, W. Yang, and G. Liu, "Research on AI security enhanced encryption algorithm of autonomous IoT systems," *Inf. Sci. (Ny)*, vol. 575, pp. 379–398, 2021.

[23] Z. Sun, J. Wan, L. Yin, Z. Cao, T. Luo, and B. Wang, "A blockchain-based audit approach for encrypted data in federated learning," *Digit. Commun. Netw.*, vol. 8, no. 5, pp. 614–624, 2022.

[24] Q. Zhang, "An overview and analysis of hybrid encryption: The combination of symmetric encryption and asymmetric encryption," in *2021 2nd International Conference on Computing and Data Science (CDS)*, 2021, pp. 616–622.

[25] C. Li *et al.*, "Efficient medical big data management with keyword-searchable encryption in healthchain," *IEEE Syst. J.*, vol. 16, no. 4, pp. 5521–5532, 2022.

[26] Y. Sun, X. Li, F. Lv, and B. Hu, "Research on logistics information blockchain data query algorithm based on searchable encryption," *IEEE Access*, vol. 9, pp. 20968–20976, 2021.

[27] A. Nadeem and M. Y. Javed, "A performance comparison of data encryption algorithms," in *2005 International Conference on Information and Communication Technologies*, 2006, pp. 84–89.

[28] Y. Chen, B. Hu, H. Yu, Z. Duan, and J. Huang, "A threshold proxy re-encryption scheme for secure IoT data sharing based on blockchain," *Electronics (Basel)*, vol. 10, no. 19, p. 2359, 2021.

[29] N. Anciaux, L. Bouganim, and Y. Guo, "Database encryption," in *Encyclopedia of Cryptography, Security and Privacy*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2023, pp. 1–6.

[30] T. M. Zaw, M. Thant, and S. V. Bezzateev, “Database security with AES encryption, elliptic curve encryption and signature,” in *2019 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF)*, 2019, pp. 1–6.

[31] P. Grubbs, T. Ristenpart, and V. Shmatikov, “Why your encrypted database is not secure,” in *Proceedings of the 16th Workshop on Hot Topics in Operating Systems*, 2017, pp. 162–168.

[32] M. Ocenas, I. Homoliak, P. Hanacek, and K. Malinka, “Security and encryption at modern databases,” in *Proceedings of the 2020 4th International Conference on Cryptography, Security and Privacy*, 2020, pp. 19–23.

[33] P. Matta, M. Arora, and D. Sharma, “A comparative survey on data encryption Techniques: Big data perspective,” *Mater. Today*, vol. 46, pp. 11035–11039, 2021.

[34] M. Sourav, “Popular SQL Server database encryption choices,” *arXiv [cs.DB]*, 2019.

[35] S. Ma, Y. Mu, and W. Susilo, “A Generic Scheme of plaintext-checkable database encryption,” *Inf. Sci. (Ny)*, vol. 429, pp. 88–101, 2018.