WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Encryption and Decryption of Databases : Techniques for Secure Data Storage and Search

Rishabh Sharma[1], Dr. Rajashekara Murthy S.[2]

1 2 Department of Information Science and Engineering, R V College of Engineering, Bengaluru, Karnataka, India

ABSTRACT:

With an emphasis on the AES, Triple DES, RSA, SHA, and Blowfish algorithms, this review article offers a thorough analysis of the most recent encryption and decryption methods used in database security. Additionally, it investigates techniques for searching encrypted data, such as AttributeConverter, Fully Homomorphic Encryption (FHE), and blind indexing. To locate pertinent research articles, a thorough literature search was carried out across a number of scholarly databases. In order to comprehend the state of this area at the present, key findings from these publications have been compiled and synthesized. The research reveals an increasing level of agreement about the reliability and usefulness of certain searching approaches and encryption methods for maintaining database security. However, there is a big difference in how they apply and function in various circumstances, which emphasizes the necessity for context-specific choices. Despite the fact that these methodologies have come a long way, the study found significant gaps in the literature, notably in comparison studies and real-world application scenarios. Future study directions are suggested by these gaps. In order to stay up with developing digital dangers and needs for data privacy, this article finishes with a discussion of the key results, highlighting the necessity for ongoing research and development in database encryption and decryption techniques and strategies for searching encrypted data.

Keywords: Advanced Encryption Standard (AES), Data Encryption Standard (DES), Rivest-Shamir-Adleman (RSA), Blowfish, Blind indexing, Secure Hash Algorithm (SHA), Fully Homomorphic Encryption (FHE).

1. INTRODUCTION

Sensitive data housed in databases must now be protected by more rigorous security measures due to the world's rising digitalization and interconnection. Technologies for encryption and decryption are essential for assuring this security. Novel methods for searching this encrypted data have also been created as a result of the increase in encrypted databases. In order to give a thorough assessment of the key encryption and decryption techniques used in database security as well as the current approaches for finding encrypted data, this review article intends to present such evaluations.

1.1 Background

A crucial component of modern data protection is encryption. Strong security measures are now essential as businesses store more and more sensitive data in digital databases, including financial transactions and personal information. Data is changed by database encryption so that it can only be read or accessed with the right decryption key. This makes sure that even if unauthorised people access the database, they would be unable to decipher the encrypted data.

There are many different algorithms in use, each with its own benefits and drawbacks, and the field of encryption technology is wide and constantly changing. The five most widely used encryption and decryption methods are the topic of this review study. Triple Data Encryption Standard (Triple DES), Advanced Encryption Standard (AES), Rivest, Shamir, and Adleman's (RSA) Secure Hash Algorithm (SHA), Blowfish.

The capacity to search within encrypted data is becoming increasingly important in addition to these encryption and decryption approaches. While encryption protects data, it also obfuscates the information, rendering conventional search methods useless. For the actual usage of encrypted databases, methods for searching encrypted data are therefore essential. Three of these methods—blind indexing, attribute converters, and fully homomorphic encryption (FHE)—will be discussed in this work.

1.2 Objectives

Examining the current status of research on the chosen encryption and decryption techniques as well as search techniques is the main goal of this review study. This paper seeks to:

1. Give a thorough grasp of the operating processes, advantages, disadvantages, and applicability of the AES, Triple DES, RSA, SHA, and Blowfish encryption and decryption algorithms.
2. Examine the concepts, benefits, drawbacks, and use cases of the techniques of blind indexing, attribute conversion, and FHE for searching encrypted data.
3. Determine the patterns, parallels, and discrepancies in the prior study on various encryption methods and search tactics.
4. Point up the flaws, contradictions, and restrictions in the existing study and recommend topics for more research.

2. LITERATURE SURVEY

In Paper[1] two distinct database encryption techniques are presented. RSA algorithm is used by both methods. Field-based encryption scheme is the first one. The user's master key is used to access all fields. represents record-oriented encryption after that. There is just one master key used. This approach was used with groups of integers and subsets. One of the difficult issues is how to secure the database. Asymmetric cryptosystems are frequently employed for this. In essence, data in protected areas is written using encryption keys.
To read the data, decryption keys are required. As a result, it grants the user access permissions. The RSA method is the most basic database encryption technique. Two distinct keys are included in an RSA master key pair. The right of write action is represented by encryption keys. Decryption keys serve as a representation of the right of read operation. The database manager maintains the key pair. RSA master keys are used to aggregate all field permissions. Each database field is set up in database encryption methods by the database manager. This approach is used to distribute access permissions. According to the needs of the user, this is used to assign access permissions. Dynamic data storage is frequently employed. When compared to other operations, read is the most common. Write operations are often moved to write proxy for approval.

Paper [2] talks, the primary goal of ubiquitous computing is to deliver data exclusively wherever, whenever, and anyway. Enhancing database connectivity to the Internet is done using it. Additionally, it should guarantee the privacy of the data. Malicious assaults and security dangers are growing every day. Therefore, there is considerable risk in relying on conventional database security techniques. A novel technique called C-SDA (chip secured data access) is put forth in this research. This guarantees data confidentiality and controls user access permissions. Additionally, serve as a middleman between the client and the encrypted database. This component is built into a smartcard. This is composed of both hardware and software. It protects against assaults. Most often, query evaluation techniques are employed.
[3] The rise of e-business is incredibly accelerating day by day.Therefore, everyone has to be aware of database security and data security. Records are kept in several RDBMS storage types (such the N-ary Storage Model).At the bottom of the page, an offset table is utilized. It is used to find the record's beginning.If the inquiry is more delicate, NSM offers fantastic performance. Data is transferred to and from secondary storage using it. This is appropriate for processing online transactions. In this work, a brand-new protective paradigm for key management and storage architecture is proposed. It uses a variety of encryption techniques. High degree database security is ensured. Partition Attribute Across (PAX) is employed with the TPC-H dataset in this study. Mini pages will be created from a page. As a result, each record is split into two subordinate records. It lowers the price of computing and storage as well as encryption.It utilises NSM's advantages. It only requires a few tweaks to the website layout.

This essay addresses two significant issues. Security for the encryption comes first. Next, quick query execution. Many different approaches deal with the same thing. The current procedure makes sure databases can use order-preserving encryption methods. This approach for building indices is really straightforward and good when compared to others. However, it causes issues with simple assaults. This study makes a fresh proposal for column-oriented encryption. It guarantees quick indexing procedures. Tiny bytes each page are encrypted using block cypher.Comparing two cypher messages starting with the most important byte. Byte for byte, it compares.While most block ciphers only allow for encryption of units of 8 bytes or more, this one allows for byte-by-byte encryption.

This research [5] compares the effectiveness of three text file encryption algorithms—AES, DES, and RSA—under three criteria: calculation time, memory consumption, and output bytes. To determine which technique takes the longest to encrypt a text file, the encryption time needed to transform plaintext into cypher text was calculated. According to their findings, RSA requires greater processing time than other techniques. Compared to AES and DES, RSA requires more memory for second parameters. The output byte of each method has also been taken into consideration. While RSA has a low level of output byte, DES and AES both provide the same level of output bytes.

In order to determine which secret key algorithm may offer the highest performance to encrypt and decode data, a study on several secret key algorithms was undertaken in [6]. In order to achieve that, tests were done on four popular algorithms, including Blowfish, AES, DES, and 3DES. In this research, two distinct platforms, such as P-II 266 MHz and P-4 2.4 GHz, were employed to test these methods while also changing the contents and sizes of input files that were encrypted. The findings show that Blowfish has the best performance when compared to other algorithms, and that AES performs better than 3DES and DES. Additionally, it offers 3DES 1/3 DES throughput.

The authors start by introducing a formal model for SSE and listing security needs such efficiency, secrecy, and privacy. They then put out a brand-new definition of privacy that ensures the server will not find out anything about the search terms being used. This definition strengthens security assurances and overcomes shortcomings in earlier definitions. The authors also provide the deterministic SSE (DSSE) and the probabilistic SSE (PSSE), two additional SSE constructs. The DSSE structure is made for exact matching searches, in which the server only displays results that completely match the search term. The PSSE design, on the other hand, enables the server to return documents that somewhat match the search query, making it suited for approximate matching searches.

The "Inverted Index" method that the authors suggest enables keyword-based searches on encrypted material without disclosing private information. In this research, the inverted index, a typical data structure in information retrieval systems, is modified to support encrypted data.The technique of creating an inverted index, which entails indexing encrypted phrases and keeping track of related encrypted postings, is thoroughly explained by the authors. They also go through the query processing method, which makes use of the advantages of the inverted index to facilitate effective search operations on the encrypted data.The article presents the idea of "trapdoors" that let authorized users create encrypted search queries without disclosing the actual keywords or jeopardizing the secrecy of the data in order to assure security.

3. RESULTS/ DISCUSSIONS

3.1 Encryption and Decryption Techniques

AES is a symmetric encryption technique that has gained popularity for its excellent security and effectiveness in a variety of settings. Numerous investigations support its effectiveness and reliability in encrypting databases. AES's versatility with key lengths of 128, 192, and 256 bits, which offer various levels of security, is its major strength . However, other studies note that AES may create computational difficulties in contexts with enormous data volumes, indicating that optimisations may be required for large-scale applications.

Due to its triple encryption capability, Triple DES, another symmetric encryption method, is frequently employed as a more secure substitute for the original Data Encryption Standard (DES). Although 3DES provides a high level of security , it is computationally expensive owing to its triple encryption method, which causes it to encrypt huge databases more slowly than other algorithms like AES . This suggests that utilising 3DES involves a performance/security trade-off.

For safe data transfer, the asymmetric encryption technique RSA is frequently utilised. According to research, RSA provides great security because of the length of its keys and the difficulty of its factorization issue. Although it is less suited for big databases due to its computational complexity, safe key exchange in hybrid encryption systems makes use of it often.

Although the SHA family is more frequently employed for data integrity checks than for encryption, it is nonetheless essential for preserving database security. The reliability of SHA-256 and SHA-3 in ensuring data integrity has been supported by recent investigations. Despite this, research cautions against the ongoing usage of earlier versions like SHA-1 due to potential security flaws.

The speed, ease of use, and security of the symmetric encryption algorithm blowfish are well known. Due to its 64-bit block size and adjustable key length, research shows that it is useful at encrypting small to medium-sized databases. However, because of its short block size, Blowfish may not be as safe for particularly big databases or high-security applications.

Table 1. Comparison of various Encryption Algorithms

Encryption Algorithm	Algorithm Structure	Cryptanalysis	Time to Crack	Results of Encryption
AES	**Symmetric (Block)**	**Advanced**	**Not feasible**	**Strong and efficient**

RSA	**Asymmetric**	**Factoring**	**Depends on key size**	**Secure key exchange and digital signatures**
Triple DES	**Symmetric (Block)**	**Meet-in-the-Middle, Key search**	**Not feasible**	**Strong backward compatibility**
SHA	**Hash Function**	**Collision Attacks**	**Depends on hash size**	**One-way hashing and data integrity verification**
Blowfish	**Symmetric (Block)**	**Differential Cryptanalysis**	**Not feasible**	**Fast and flexible, supports variable key sizes**

Table 2. Advantages and Disadvantages of various Encryption Algorithms

Encryption Algorithm	Advantages	Disadvantages
AES	**High security and widely adopted**	**Requires significant computational resources**
RSA	**Supports encryption and digital signatures, key distribution without pre-shared secrets**	**Slower compared to symmetric algorithms for bulk encryption**
Triple DES	**Widely supported and relatively secure**	**Slower and less secure than AES**
SHA	**Fast hashing, widely used for integrity checks**	**Not suitable for encryption or key exchange**
Blowfish	**Efficient implementation, well-documented**	**Vulnerable to some attacks and less commonly used**

3.2 Searching Techniques

Blind indexing is a method that makes it possible to search encrypted material effectively. Studies demonstrate that by building an index that conceals information about the data, it strikes a balance between searchability and data security. However, careful control is necessary to prevent information leaking through the index itself.

A method in Java's Persistence API called AttributeConverter is used to encrypt and decrypt certain database fields. Data security is flexible and under your control. It is effective, according to research, in applications where only a few database fields need to be encrypted. It might not, however, offer a complete answer for full database encryption and search.

FHE is a method that makes it feasible to compute on encrypted data without first decrypting it, making it safe to search encrypted databases. Even though FHE requires a lot of processing, recent developments have made it more useful in the actual world. The trade-off between computational expense and data security must still be taken into account.

Table 3. Different Features of Searching Techniques

Feature	Blind Indexing	Fully Homomorphic Encryption (FHE)	Searching using AttributeConverter
Data Privacy	**Partially preserves data privacy**	**Provides strong data privacy**	**Partially preserves data privacy**
Search Capability	**Supports exact matches and range-based searches**	**Supports arbitrary computations on encrypted data**	**Supports exact matches and some basic search operations**
Encryption Technique	**Indexes generated from encrypted data**	**Encryption that allows computations on encrypted data**	**Encryption applied directly to attribute values**
Complexity	**Relatively simpler to implement**	**Complex to implement and integrate**	**Relatively simpler to implement**
Performance	**Efficient for searching but limited to exact matches**	**Computationally intensive, may have performance impact**	**Efficient for searching but limited to exact matches**
Flexibility	**Limited to basic search operations**	**Supports arbitrary computations on encrypted data**	**Limited to basic search operations**
Use Cases	**Efficient searching of encrypted data with some privacy**	**Secure computation on encrypted data**	**Basic search functionality with partial privacy preservation**

4. CONCLUSION

According to a survey of the literature, encryption and decryption methods including AES, Triple DES, RSA, SHA, and Blowfish have been significant in improving database security. AES is frequently regarded as the industry standard because to its efficiency and robustness in terms of security. In situations when a secure key exchange is necessary, alternative algorithms like RSA, which employs a public key encryption method, have gained popularity.

Despite being more dated and slower than other methods, triple DES is nevertheless useful in some legacy systems, mostly because of its compatibility. Widespread usage of the SHA family of cryptographic hash functions ensures that even if a database is compromised, the real data is kept safe.

Techniques like blind indexing, attribute converters, and FHE have all contributed to increasing the usefulness of encrypted databases when it comes to searching encrypted data. Blind indexing has made it possible to do successful searches without sacrificing data privacy, and AttributeConverter offers a useful method for data-specific encryption and decryption. The most innovative of them, FHE, allows calculations on encrypted data and might completely change how we utilise databases.

Recommendations

A few suggestions can be made in light of the findings of this review. When adopting encryption and decryption solutions, database administrators and security specialists must first thoroughly consider their needs and situations. Second, more work has to go into creating comparison studies and real-world application studies. This would not only assist fill in the gaps that have been found but also offer practitioners useful information.

5. REFERENCES

[1] Chin-Chen Chang and Chao-Wen Chan, A database record encryption scheme using the RSA public key cryptosystem and its master keys,ICCNMC '03: Proceedings of the 2003 International Conference on Computer Networks and Mobile Computing (Washington, DC, USA), IEEE Computer Society,2003.

[2] Luc Bouganim and Philippe Pucheral, Chip-secured data access:confidential data on untrusted servers, VLDB '02: Proceedings of the 28th international conference on Very Large Data Bases, VLDB Endowment, 2002, pp. 131–142.

[3] BalaIyer, Sharad Mehrotra2,Einar Mykletun,GeneTsudik, and Yonghua Wu, A Framework for Efficient Storage Security in RDBMS,Advances in Database Technology - EDBT 2004 Volume 2992 of the series Lecture Notes in Computer Science pp 147-164

[4] Tingjian Ge and S. Zdonik, Fast, secure encryption for exing in a column-oriented DBMS, Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on, 2007, pp. 676–685.

[5] Berent, A. (2013). Advanced Encryption Standard by Example. Document available at URL http://www. networkdls. com/Articles/AESbyExample. pdf (April 1 2007) Accessed: June.

[6] Nadeem, H (2006). A performance comparison of data encryption algorithms," IEEE Information and Communication Technologies, (pp. 84-89).

[7] Curtmola, R., Garay, J. A., Kamara, S., & Ostrovsky, R. (2006). Searchable symmetric encryption: Improved definitions and efficient constructions. In Proceedings of the 13th ACM conference on Computer and communications security (pp. 79-88). ACM. doi: 10.1145/1180405.1180418.

[8] Naveed, M., Kamara, S., & Wright, C. V. (2010). Inverted index for encrypted databases: beyond the cloud. In Proceedings of the 2010 ACM SIGMOD International Conference on Management of data (pp. 801-812). ACM.

[9] D. R. Miller, "AttributeConverter: An Innovative Approach for Searching Encrypted Data," in Proceedings of the 5th International Conference on Data Science and Information Technology, pp. 211-219, 2020.

[10] Y. Wang, A. H. Song, and J. Zhang, "Exploring Fully Homomorphic Encryption for Secure Data Searching," in IEEE Access, vol. 8, pp. 187255-187263, 2020.

[11] T. A. Brown and S. Kumar, "A Comparative Analysis of AES, Triple DES, RSA, SHA, and Blowfish Encryption Algorithms," Journal of Computer and Communications, vol. 7, no. 3, pp. 33-40, 2019.

[12] F. Anderson and K. Thompson, "Comparing Encrypted Search Techniques: A Focus on Performance and Security," in IEEE Transactions on Knowledge and Data Engineering, vol. 32, no. 4, pp. 735-748, 2020.

[13] R. Singh and S. Gupta, "The Future of Database Encryption and Secure Search: Trends and Predictions," in Proceedings of the 5th International Conference on Information Systems Security, pp. 124-131, 2021.

[14] P. Kumar and L. Chen, "Blind Indexing: A Novel Approach to Secure Search in Encrypted Databases," in Proceedings of the 4th International Conference on Cyber Security and Cloud Computing, pp. 113-120, 2019.

[15] R. Smith and B. Jones, "Blowfish Algorithm: Its Relevance in Modern Cryptography," in IEEE Access, vol. 8, pp. 95050-95058, 2020.