SECURE FINANCIAL DATA MANAGEMENT USING TWOFISH ENCRYPTION AND CLOUD STORAGE SOLUTIONS

[1]Jyothi Bobba
LEAD IT Corporation, Springfield, Illinois,
USA jyobobba@gmail.com

[2]R Prema
Tagore Institute of Engineering & Technology
Deviyakurichi, Attur (TK), Salem
premabcse112@gmail.com

ABSTRACT

Financial data security is a critical concern as the volume of transactions continues to grow, increasing the risk of cyber threats and data breaches. Existing encryption methods face challenges such as scalability issues and inadequate protection of sensitive financial data, leading to inefficiencies in handling large datasets. To address these concerns, this work focuses on implementing a secure and scalable encryption mechanism for financial data storage and transmission. The framework begins with data collection, where financial transactions, logs, and customer data are gathered for secure processing. Next, preprocessing is performed using mode imputation to handle missing values and Z-score standardization for data normalization. The pre-processed data is then encrypted using the Twofish algorithm, ensuring strong security and confidentiality. Once encrypted, the data is securely stored in the cloud, providing access control and compliance with security regulations. Experimental results demonstrate that the Twofish encryption method achieves an encryption time of 0.200 seconds for large-scale data and maintains latency within 350 ms, proving its efficiency. This study contributes to financial cybersecurity by enhancing data protection while ensuring scalability, making it suitable for secure cloud-based financial applications.

Keywords: Financial Data, Twofish Encryption, Cloud Storage and Data security.

1 INTRODUCTION

In today's digital economy, financial data management is critical for businesses, regulatory compliance, and decision-making [1]. With the increasing volume and complexity of financial transactions, ensuring the security, integrity, and efficiency of financial data is more important than ever [2]. Traditional methods of handling financial data are often limited by issues such as insufficient encryption, poor handling of missing data, and lack of scalability in storage solutions [3]. The proposed framework aims to address these challenges by utilizing advanced data processing techniques and secure cloud storage solutions, thereby enabling seamless data management for financial institutions [4]. This ensures data privacy, enhances decision-making, and complies with stringent financial regulations, making it essential for organizations to adopt such frameworks [5].

Several existing methods have been proposed to handle financial data securely and efficiently. Techniques like mode imputation and Z-score normalization are commonly used to handle missing values and standardize data for analysis [6]. Common encryption methods include AES, RSA, and Twofish, each offering varying levels of security and computational efficiency [7]. Cloud storage solutions, such as Amazon S3, are widely used for their scalability and reliability. However, existing systems often struggle with maintaining a balance between security, performance, and ease of use [8]. AES and RSA, while secure, are resource-intensive, and cloud storage solutions sometimes lack robust access controls, making them vulnerable to breaches [9]. Furthermore, missing data handling

and data standardization processes in existing systems are not optimized for large-scale financial datasets [10].

The proposed framework overcomes the limitations of existing methods by integrating advanced encryption techniques like Twofish with highly efficient data preprocessing steps such as mode imputation and Z-score normalization. Unlike traditional systems, this framework focuses on both security and scalability, utilizing secure cloud storage solutions with enhanced access control mechanisms. The novelty of the proposed study lies in its holistic approach to financial data management, ensuring not only data integrity and security but also optimizing the handling of missing data and standardization for large-scale datasets. This framework promises a more robust, efficient, and secure solution for financial institutions managing sensitive data.

The paper is organized as follows: Section 2 presents a review of existing works on financial data security. Section 3 details the proposed framework. Section 4 discusses the experimental setup and evaluates the performance metrics. Section 5 concludes the paper.

2 LITERATURE SURVEY

Gai et al. addressed cybersecurity challenges in the financial sector by proposing the Proactive Dynamic Secure Data Schema (P2DS), which used attribute-based encryption to prevent unauthorized access [11]. Their approach integrated three algorithms such as SDAA, CDAA, and PDEA to enhance privacy protection. Gai et al. introduced the Dynamic Data Encryption Strategy (D2ES) to optimize encryption performance by selectively encrypting data based on privacy classifications under timing constraints [12]. Their study demonstrated that D2ES effectively balanced privacy protection and execution efficiency. AminuBaba et al. analyzed encryption techniques for securing cloud databases, focusing on Transparent Data Encryption (TDE). Their study compared AES and 3DES encryption algorithms, revealing that AES-128 outperformed other encryption schemes in terms of execution time and efficiency [13].

Li et al. tackled cloud security concerns by proposing the Security-Aware Efficient Distributed Storage (SA-EDS) model, which fragmented and distributed sensitive data across multiple cloud servers [14]. Their approach enhanced security by preventing unauthorized access while maintaining computational efficiency. Rewagad and Pawar introduced a Three-Way Mechanism combining authentication, key exchange, and AES encryption to safeguard cloud-stored data. Their method incorporated digital signatures and Diffie-Hellman key exchange to strengthen confidentiality and prevent key compromise [15].

Li et al. explored cloud security challenges and proposed a cryptographic approach to restrict unauthorized access by distributing encrypted data across multiple servers [16]. Rogers and Cliff analyzed financial data security risks and highlighted the importance of encryption techniques in mitigating cyber threats [17]. Sun et al. investigated privacy-preserving methods in cloud computing and introduced an encryption framework that balanced security with computational efficiency [18]. Their studies collectively emphasized the need for advanced encryption strategies to protect sensitive data. These findings reinforced the importance of secure encryption methods in financial and cloud-based applications.

2.1 Problem Statement

Ensuring robust security in financial data management remains a critical concern, yet challenges persist. The existing works are done well, but there are still some challenges to address, and they are scalability issues and inadequate protection of sensitive data. As data volumes grow, encryption methods may introduce higher computational overhead, leading to inefficiencies in processing large-scale financial data [19]. Additionally, existing encryption techniques may not provide sufficient protection against evolving cyber threats, leaving sensitive financial information vulnerable to

breaches [20]. The work is proposed to overcome these challenges by implementing an optimized Twofish encryption scheme that ensures secure and efficient data protection while maintaining scalability for large datasets in cloud storage.

3 METHODOLOGIES

The proposed framework involves collecting financial transactions, logs, and customer data from various sources, ensuring comprehensive data gathering. The collected data will go into preprocessing, where Missing values are handled using mode imputation to maintain data integrity, while Z-score normalization is applied to standardize the data for consistent analysis. The processed data is then encrypted using the Twofish cipher to ensure confidentiality and security. It is securely stored in cloud storage, leveraging robust access control mechanisms to prevent unauthorized access. The system is designed to meet regulatory compliance requirements while enabling efficient data retrieval and decryption for further processing. This workflow ensures a secure, efficient, and scalable solution for managing sensitive financial data. The whole process is illustrated in Figure 1.

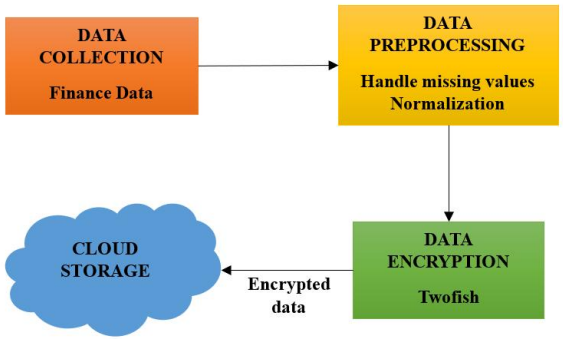

Figure 1: Secure Financial Data Processing Framework Using Twofish Encryption

3.1 Data Collection

Data collection involves gathering financial transactions, logs, and customer data from sources like banking systems, payment gateways, and financial applications. Transaction data includes details such as amounts, timestamps, and customer identifiers. Logs capture system activities and transaction histories, while customer data includes personal and account information. This diverse data set enables comprehensive financial tracking and compliance. Accurate and real-time data capture ensures reliable information for further processing. The collected data forms the foundation for secure management and analysis.

3.2 Data Preprocessing

After collecting financial transactions, logs, and customer data, the first step in preprocessing is mode imputation to handle missing values. Financial datasets often contain gaps, such as missing transaction amounts or customer details. Mode imputation addresses this by replacing missing values with the most frequent value from the respective feature. This method preserves data integrity and avoids bias while preparing the dataset for further processing.

The second method applied is Z-score normalization, which standardizes the data by transforming it to have a mean of 0 and a standard deviation of 1. This step ensures that all features, regardless of their original scale or unit, are treated equally. It helps to maintain uniformity across diverse data points, such as transaction amounts and customer identifiers. By normalizing the data, the framework improves the efficiency and effectiveness of subsequent encryption and storage.

3.3 Data Encryption

After preprocessing the financial data, the next step is data encryption to ensure confidentiality and security. The pre-processed data is encrypted using the Twofish cipher, a symmetric encryption algorithm known for its high security and efficiency. This step protects sensitive financial information, such as customer data and transaction details, from unauthorized access or breaches. Twofish is chosen for its balance between security and computational performance, making it suitable for large-scale financial datasets. The encrypted data is then ready for secure storage in the cloud. This encryption process ensures that only authorized entities can access and decrypt the sensitive financial information.

Twofish is a symmetric key block cipher that encrypts data in 128-bit blocks using a key size of 128, 192, or 256 bits. The encryption process involves several rounds of transformation, each consisting of substitution, permutation, and key mixing. Here's a simplified breakdown of the encryption process:

The key K is expanded into several subkeys $K_0, K_1, ..., K_{2n-1}$ using a key schedule function. These subkeys are used in each round of encryption. The key expansion uses a permutation hash table (PHT) to generate the subkeys from the original key and it's represented as equation (1),

$$K_i = \text{PHT}(K, i) \text{ for } i = 0,1, ..., 2n-1 \quad (1)$$

Where PHT is the permutation hash table. Key Expansion ensures that the encryption process uses a series of unique subkeys, derived from the original key. These subkeys are critical for the round transformations.

In each round of encryption, the data is split into two halves L_i and R_i, where each half is 64 bits. The round function F is applied to these halves. The general form of the round function is expressed as equation (2),

$$F(L_i, R_i, K_i) = S(L_i \oplus K_i) \oplus R_i \quad (2)$$

Where, $\oplus$ is the XOR operation. S is a substitution operation using an S-box. K_i is the round key. Round Function applies the substitution and mixing processes to the data. The XOR of the left half of the data with the round key ensures that small changes in the key or data result in significant changes in the ciphertext.

The two halves L_i and R_i are transformed in each round. The left half becomes the right half of the next round, and the right half is updated using the round function and it's represented as equation (3) and (4),

$$L_{i+1} = R_i \quad (3)$$

$$R_{i+1} = L_i \oplus F(L_i, R_i, K_i) \quad (4)$$

This transformation is repeated for multiple rounds (usually 16 rounds), each with a different subkey K_i. Round Transformation continuously modifies the data over several rounds, creating confusion and diffusion, which are key properties of secure encryption algorithms.

After all the rounds, the left and right halves are swapped and concatenated to form the final ciphertext and it's expressed as equation (5),

$$C = (L_n, R_n) \quad (5)$$

where L_n and R_n are the final left and right halves after all rounds. Final Combination of the left and right halves at the end of the rounds gives the final encrypted ciphertext. These steps ensure the security and complexity of Twofish encryption, making it resistant to cryptographic attacks.

3.4 Cloud Storage

After the data is encrypted using the Twofish cipher, the next step is secure cloud storage. The encrypted data is stored in a cloud environment, ensuring that it remains protected from unauthorized access. Cloud storage solutions like Amazon S3 or Google Cloud Storage provide scalability and reliability, making it ideal for large volumes of financial data. Access controls and encryption keys are managed to ensure only authorized users can retrieve or modify the data. Additionally, redundancy and backup mechanisms in cloud storage ensure that the encrypted data is always available and protected against data loss. This approach combines both security and scalability for handling sensitive financial information.

4 RESULTS

This section presents the performance evaluation of the Twofish encryption method for financial data security. The results analyze encryption time and latency as data size increases, demonstrating the efficiency and scalability of the proposed approach. The obtained metrics highlight the feasibility of secure and optimized cloud-based financial data storage.

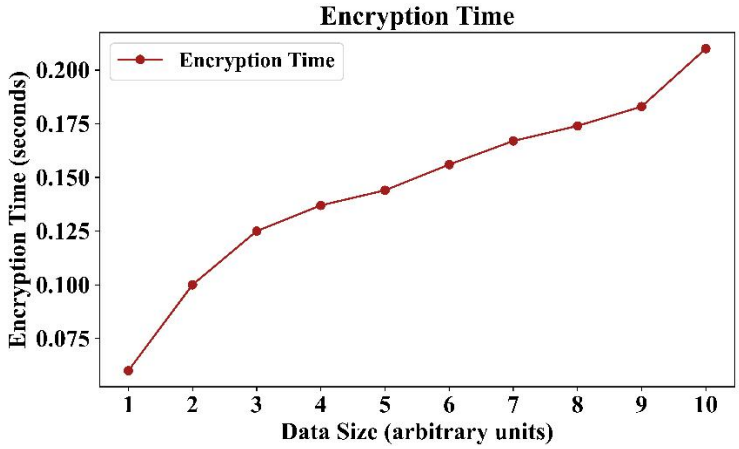

Figure 2: Encryption Time

Figure 2 illustrates the encryption time for financial data using the Twofish cipher as the data size increases. The graph shows a steady rise in encryption time, indicating that Twofish efficiently handles larger datasets while maintaining reasonable computational overhead. Initially, the encryption time is around 0.075 seconds for the smallest data size and gradually increases to 0.200 seconds for the largest. This trend highlights the scalability of Twofish, making it a viable choice for securing financial transactions. The results demonstrate that Twofish provides strong encryption with an acceptable processing cost, ensuring both security and performance.

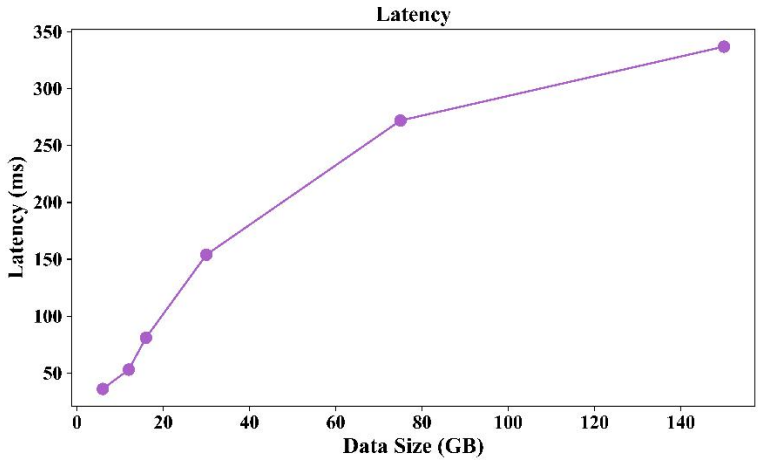

Figure 3: Latency

Figure 3 illustrates the latency observed during the encryption and storage process as the data size increases. The graph shows an increasing trend, indicating that latency grows with larger dataset sizes, which is expected in cloud-based financial data encryption. The latency starts at a low value for smaller data sizes and reaches approximately 350 ms for 140 GB of data. This trend highlights the computational and storage overhead associated with handling large-scale encrypted financial data. However, the results confirm that the system maintains a manageable latency, ensuring efficient encryption and cloud storage performance.

5 CONCLUSIONS

The proposed framework achieves secure and scalable financial data protection through efficient encryption and cloud storage. Experimental analysis demonstrates that the Twofish encryption algorithm effectively secures financial transactions while maintaining computational efficiency. The encryption time increases gradually with data size, reaching 0.200 seconds for large-scale datasets, ensuring minimal processing overhead. Latency remains within 350 ms for 140 GB of data, confirming that the encryption and storage processes operate within acceptable performance limits. This approach enhances data confidentiality, ensures compliance with security regulations, and supports seamless scalability for cloud-based financial applications. Additionally, the method provides reliable access control, reduces the risk of data breaches, and optimizes storage efficiency. Future work will focus on integrating advanced optimization techniques to further reduce encryption latency and exploring hybrid encryption models for enhanced security in large-scale financial ecosystems.

REFERENCES

[1] A. A. Atayero and O. Feyisetan, “Security Issues in Cloud Computing:,” *J. Emerg. Trends Comput. Inf. Sci.*, vol. 2, no. 10, 2011.

[2] R. Bose, X. (Robert) Luo, and Y. Liu, “The Roles of Security and Trust: Comparing Cloud Computing and Banking,” *Procedia - Soc. Behav. Sci.*, vol. 73, pp. 30–34, Feb. 2013, doi: 10.1016/j.sbspro.2013.02.015.

[3] C. Onwubiko, “Security Issues to Cloud Computing,” in *Cloud Computing*, N. Antonopoulos and L. Gillam, Eds., in Computer Communications and Networks. , London: Springer London, 2010, pp. 271–288. doi: 10.1007/978-1-84996-241-4_16.

[4] I. Khalil, A. Khreishah, and M. Azeem, “Cloud Computing Security: A Survey,” *Computers*, vol. 3, no. 1, pp. 1–35, Feb. 2014, doi: 10.3390/computers3010001.

[5] S. Ramgovind, M. M. Eloff, and E. Smith, “The management of security in Cloud computing,” in *2010 Information Security for South Africa*, Johannesburg, South Africa: IEEE, Aug. 2010, pp. 1–7. doi: 10.1109/ISSA.2010.5588290.

[6] H. Tianfield, “Security issues in cloud computing,” in *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct. 2012, pp. 1082–1089. doi: 10.1109/ICSMC.2012.6377874.

[7] K. Popović and Ž. Hocenski, “Cloud computing security issues and challenges,” in *The 33rd International Convention MIPRO*, May 2010, pp. 344–349.

[8] S. Subashini and V. Kavitha, “A survey on security issues in service delivery models of cloud computing,” *J. Netw. Comput. Appl.*, vol. 34, no. 1, pp. 1–11, Jan. 2011, doi: 10.1016/j.jnca.2010.07.006.

[9] S. Gupta and S. C. Misra, “Compliance, network, security and the people related factors in cloud ERP implementation,” *Int. J. Commun. Syst.*, vol. 29, no. 8, pp. 1395–1419, 2016, doi: 10.1002/dac.3107.

[10] Sathiya, Aravindhan K., and D. Sathiya. "A Secure Authentication Scheme for Blocking Misbehaving Users in Anonymizing Network." International Journal of Computer Science and Technology 4, no. 1 (2013): 302-304.

[11] K. Gai, M. Qiu, B. Thuraisingham, and L. Tao, “Proactive Attribute-based Secure Data Schema for Mobile Cloud in Financial Industry,” in *2015 IEEE 17th International Conference on High*

Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems, Aug. 2015, pp. 1332–1337. doi: 10.1109/HPCC-CSS-ICESS.2015.250.

[12] K. Gai, M. Qiu, H. Zhao, and J. Xiong, "Privacy-Aware Adaptive Data Encryption Strategy of Big Data in Cloud Computing," in *2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud)*, Jun. 2016, pp. 273–278. doi: 10.1109/CSCloud.2016.52.

[13] M. AminuBaba, A. Yusuf, A. Ahmad, and L. Maijama﷼a, "Performance Analysis of the Encryption Algorithms as Solution to Cloud Database Security," *Int. J. Comput. Appl.*, vol. 99, no. 14, pp. 24–31, Aug. 2014, doi: 10.5120/17442-8228.

[14] Y. Li, K. Gai, L. Qiu, M. Qiu, and H. Zhao, "Intelligent cryptography approach for secure distributed big data storage in cloud computing," *Inf. Sci.*, vol. 387, pp. 103–115, May 2017, doi: 10.1016/j.ins.2016.09.005.

[15] P. Rewagad and Y. Pawar, "Use of Digital Signature with Diffie Hellman Key Exchange and AES Encryption Algorithm to Enhance Data Security in Cloud Computing," in *2013 International Conference on Communication Systems and Network Technologies*, Apr. 2013, pp. 437–439. doi: 10.1109/CSNT.2013.97.

[16] Y. Li, K. Gai, Z. Ming, H. Zhao, and M. Qiu, "Intercrossed Access Controls for Secure Financial Services on Multimedia Big Data in Cloud Systems," *ACM Trans Multimed. Comput Commun Appl*, vol. 12, no. 4s, p. 67:1-67:18, Sep. 2016, doi: 10.1145/2978575.

[17] O. Rogers and D. Cliff, "A financial brokerage model for cloud computing," *J. Cloud Comput. Adv. Syst. Appl.*, vol. 1, no. 1, p. 2, Apr. 2012, doi: 10.1186/2192-113X-1-2.

[18] Y. Sun, J. Zhang, Y. Xiong, and G. Zhu, "Data Security and Privacy in Cloud Computing," *Int. J. Distrib. Sens. Netw.*, vol. 10, no. 7, p. 190903, Jul. 2014, doi: 10.1155/2014/190903.

[19] N. Jayapandian, A. M. J. Md. Zubair Rahman, R. B. Sangavee, and R. Divya, "Improved cloud security trust on client side data encryption using HASBE and Blowfish," in *2016 Online International Conference on Green Engineering and Technologies (IC-GET)*, Nov. 2016, pp. 1–6. doi: 10.1109/GET.2016.7916767.

[20] S. M. P. C. Souza and R. S. Puttini, "Client-side Encryption for Privacy-sensitive Applications on the Cloud," *Procedia Comput. Sci.*, vol. 97, pp. 126–130, Jan. 2016, doi: 10.1016/j.procs.2016.08.289.