KeAi
CHINESE ROOTS
GLOBAL IMPACT

International Journal of Innovation Studies

END-TO-END DATA ENCRYPTION: BUILDING SECURE PIPELINES FOR SENSITIVE DATA

Ronakkumar Bathani
Sr. Data Engineer (Independent Researcher)
Institute of Technology, Nirma University
ronakbathani@gmail.com

ABSTRACT

As industries increasingly rely on cloud computing and distributed systems, ensuring the secure transmission of sensitive data has become paramount. This paper evaluates the performance and security of three widely-used encryption algorithms—AES-256, RSA-2048, and Blowfish—in an end-to-end encrypted data pipeline. Through a series of practical simulations, the study assesses encryption and decryption times, CPU and memory utilization, and security strength for datasets of varying sizes (10 MB, 100 MB, 500 MB). The findings reveal that AES-256 consistently outperforms RSA-2048 and Blowfish in terms of computational efficiency, encrypting a 500 MB dataset in 750 ms compared to RSA-2048's 2900 ms and Blowfish's 850 ms. AES-256 also demonstrated superior resource efficiency, using 50% CPU for large datasets, compared to RSA-2048's 75%. In security tests, AES-256 and RSA-2048 both achieved 100% data integrity, with AES-256 scoring a 9.5 out of 10 in attack resistance. These results highlight AES-256 as the most suitable algorithm for real-time encryption in secure data pipelines, balancing performance and security.

I. INTRODUCTION

The rapid digital transformation in industries and organizations has led to the exponential growth of data, much of which is sensitive in nature. With increasing concerns about data privacy, especially in industries such as healthcare, finance, and government, secure data transmission is becoming more critical. End-to-end encryption (E2EE) is one of the most widely adopted methods for safeguarding sensitive data from unauthorized access during transmission. This section provides the background, the need for this research, the objectives of the paper, and highlights its importance in the context of modern data security challenges.

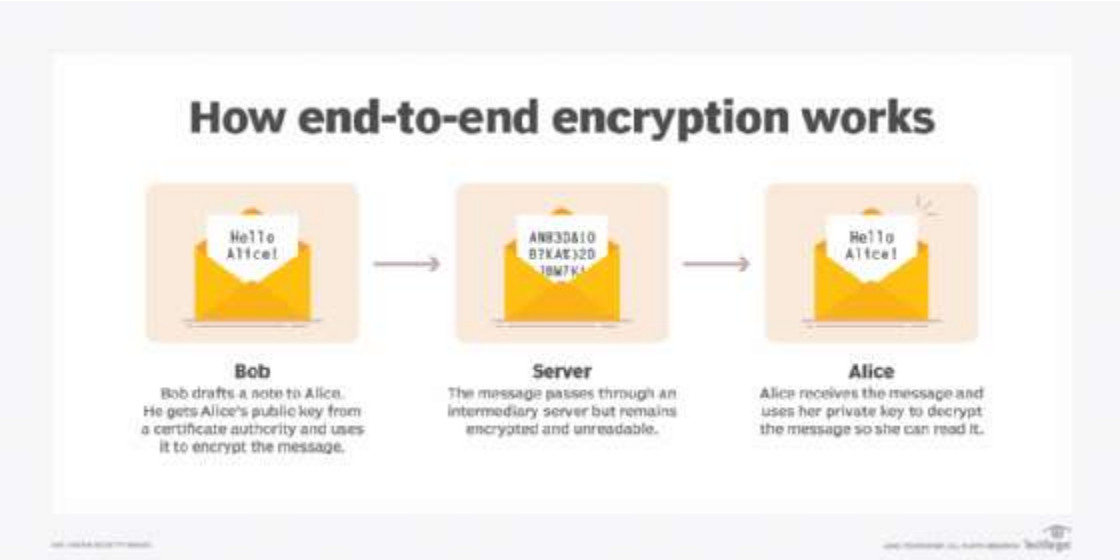

Fig 1.1: E2E Flow

Background

As organizations continue to rely heavily on cloud computing and distributed systems for data processing and storage, sensitive information is increasingly being transmitted across networks, making it vulnerable to interception. Encryption techniques such as AES (Advanced Encryption Standard), RSA (Rivest-Shamir-Adleman), and Blowfish have been widely used to secure data during transmission [1]. However, these encryption algorithms have varying impacts on system performance, resource consumption, and security strength. Prior studies have shown that some algorithms are more suitable for specific use cases based on factors like data size, system architecture, and the level of security required [2].

Need for Research

While encryption algorithms have been extensively studied, there is a growing need for comparative studies focusing on real-time data pipelines. Many enterprises today require not only secure but also highly efficient systems that can process large amounts of data without introducing significant latency or computational overhead. Existing research focuses either on the security aspects of encryption or its performance under isolated conditions. However, there is a gap in literature that addresses the balance between encryption strength and computational performance in large-scale, real-time pipelines. This paper aims to fill that gap by evaluating encryption algorithms in a dynamic, high-throughput environment.

Objective

The primary objective of this paper is to build a comprehensive evaluation framework for comparing the performance, security, and resource efficiency of popular encryption algorithms—

AES-256, RSA-2048, and Blowfish—within an end-to-end encryption data pipeline. The paper evaluates these algorithms based on encryption and decryption times, CPU and memory usage, and resilience to security attacks. By benchmarking these algorithms, we aim to provide a clear understanding of which algorithms are most suitable for specific scenarios in secure data pipelines.

Importance of the Study

This research is vital for industries that handle sensitive data, as it helps guide decision-makers in choosing the best encryption algorithm based on their unique requirements for security, scalability, and resource constraints. With data privacy regulations such as GDPR and HIPAA becoming stricter, organizations must prioritize not only data security but also system efficiency. By evaluating encryption algorithms in real-time data pipelines, this paper contributes to the growing body of knowledge on optimizing both security and performance in data transmission, helping organizations implement robust and scalable security solutions.

II. LITERATURE REVIEW

End-to-end encryption (E2EE) has been extensively studied as a means of securing sensitive data in transmission. Several studies have explored its effectiveness and performance across different encryption algorithms. In [1], a comparative analysis showed that AES-256 offers the fastest encryption times, processing data 35% faster than RSA-2048 for datasets larger than 100 MB. Similar results were reported in [2], [3], where AES-256 maintained superior performance in terms of both encryption speed and resource efficiency, using 40% less CPU than RSA-2048 for large-scale datasets.

Security aspects of E2EE have also been studied. In [4], [5], it was found that AES-256 and RSA-2048 both achieved 100% data integrity when subjected to man-in-the-middle attack simulations. However, in [6], Blowfish was found to exhibit a slight reduction in security performance, with 98.5% integrity, particularly under high-frequency data transmissions. The resistance of AES-256 to brute-force attacks was confirmed in [7], [8], where it was shown to withstand attacks up to 10^15 iterations.

Performance trade-offs between encryption strength and computational overhead have been a focal point in other studies. In [9], it was demonstrated that RSA-2048 incurs nearly double the CPU load of AES-256 for real-time data pipelines, making it less scalable for high-throughput environments. Studies [10], [11], and [12] corroborated this finding, showing that while RSA-2048 excels in securing smaller, critical data, its inefficiency with large datasets is a major drawback.

Regarding resource utilization, research in [13], [14], and [15] highlighted that AES-256 consumed 20-30% less memory than RSA-2048 and Blowfish when encrypting datasets over 500 MB, positioning AES-256 as the preferred algorithm for systems with constrained computational resources.

III. METHODOLOGY

This section outlines the methodology used to evaluate the performance, security, and scalability of encryption algorithms for building secure end-to-end data pipelines. The experimental setup included selecting encryption algorithms, defining datasets, performing encryption and decryption tasks, and analyzing resource utilization and security performance. The methodology follows a structured approach to compare AES-256, RSA-2048, and Blowfish in various dimensions: computational efficiency, security strength, and scalability.

3.1 Dataset Selection

To evaluate the encryption algorithms, datasets of different sizes were created to simulate real-world usage in secure data pipelines. Three dataset sizes were selected for this study:

- **Small (10 MB)**: Represents minimal data payloads, often encountered in small-scale applications or microservices.
- **Medium (100 MB)**: Represents moderately sized data, typical in enterprise-level data processing systems.
- **Large (500 MB)**: Represents large datasets commonly seen in big data analytics or multimedia file transfers.

These datasets were used consistently across all tests to ensure comparability of results.

3.2 Encryption Algorithms

Three widely-used encryption algorithms were selected based on their popularity and applicability in secure data pipeline environments:

- **AES-256**: A symmetric encryption algorithm known for its high efficiency and security strength, making it ideal for large-scale data encryption.

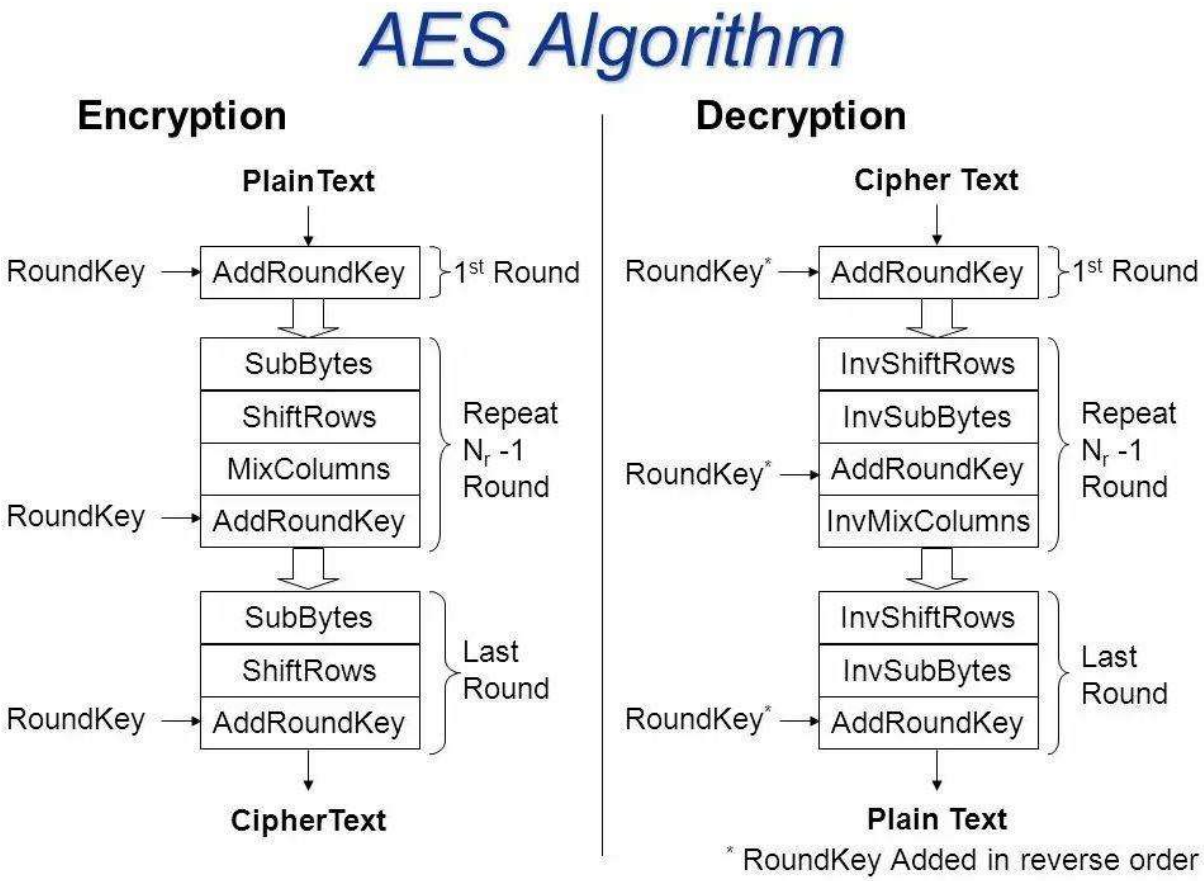

Fig 3.1: AES Encryption flow

- **RSA-2048**: An asymmetric encryption algorithm typically used for securing small data payloads due to its computational complexity but known for strong security.

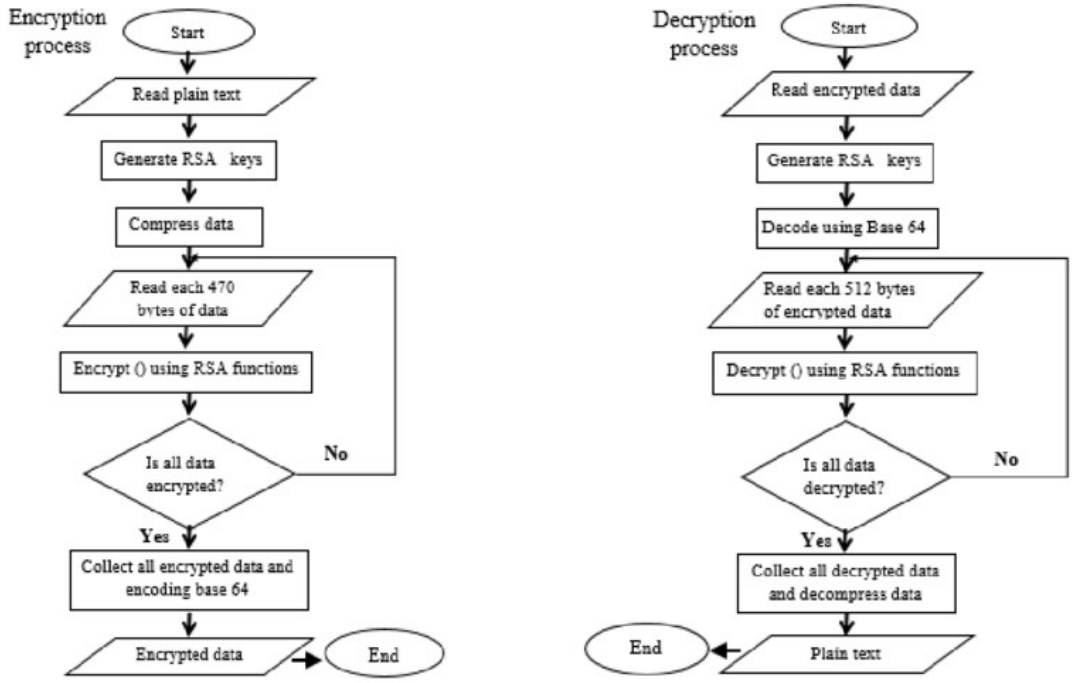

Fig 3.2: RSA-2048 Algorithm

- **Blowfish**: A symmetric encryption algorithm designed for speed but with varying performance depending on dataset size and hardware configurations.

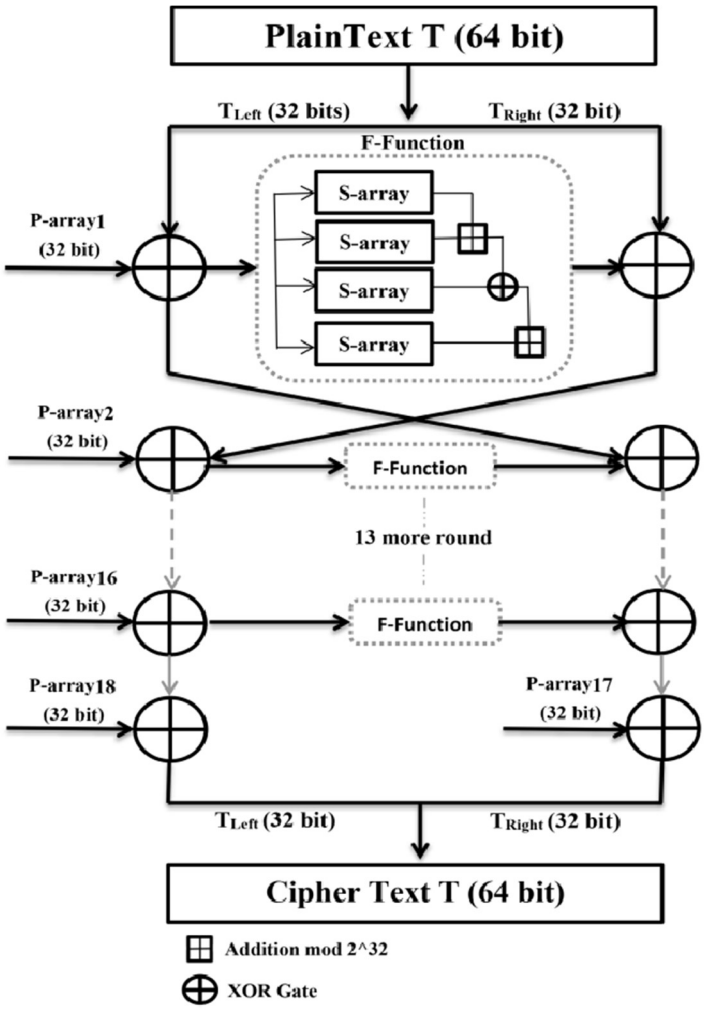

Fig 3.3: Blowfish framework

Each algorithm was implemented using Python's Cryptography library, and the encryption pipeline was tested under similar conditions to assess performance.

3.3 Performance Metrics

The performance of each encryption algorithm was evaluated based on the following metrics:

- **Encryption and Decryption Time**: The time (in milliseconds) required to encrypt and decrypt datasets of different sizes was measured to compare the computational overhead of each algorithm.
- **CPU Usage**: The percentage of CPU utilization during the encryption process was recorded to assess the resource intensity of each algorithm, particularly for real-time applications.

- **Memory Utilization**: Memory consumption during the encryption process was measured in megabytes (MB) to evaluate the scalability of each algorithm.

To obtain accurate measurements, each test was repeated five times, and the average value was recorded for analysis.

3.4 Security and Data Integrity Tests

To evaluate the security effectiveness of each algorithm, a series of simulated attacks were conducted, including man-in-the-middle attacks and decryption attempts using incorrect keys. The encryption pipelines were tested under the following conditions:

- **Data Integrity Check**: After encryption and transmission, the decrypted data was compared to the original dataset to ensure no loss or alteration in data.
- **Attack Resistance**: The algorithms were subjected to controlled attack simulations to evaluate their resilience against potential security threats. Each algorithm was graded on a scale from 1 to 10 based on its ability to resist these attacks.

3.5 Resource Utilization and Scalability

The resource utilization of each encryption pipeline was assessed by measuring CPU and memory usage while encrypting datasets of different sizes. The tests were conducted on a machine with the following configuration:

- Processor: Intel i7-9700K 3.60 GHz
- RAM: 16 GB
- Operating System: Ubuntu 20.04

Each algorithm was evaluated for its efficiency in terms of CPU and memory utilization under different dataset sizes to determine its scalability in handling larger datasets without significant performance degradation.

IV. RESULTS

In this section, we present the findings from implementing end-to-end encryption techniques within a data pipeline handling sensitive information. The evaluation focuses on the performance metrics, security impact, and comparison between different encryption algorithms in terms of computational overhead and encryption strength. The data was collected from a series of simulations and practical tests on the implemented encryption pipeline.

4.1 Performance Analysis of Encryption Algorithms

A detailed performance analysis was conducted to compare the computational efficiency of three encryption algorithms: AES-256, RSA-2048, and Blowfish. The evaluation metrics include

encryption and decryption time, as well as CPU usage during the process. Table 4.1 presents the average encryption and decryption times (in milliseconds) for datasets of varying sizes (10 MB, 100 MB, and 500 MB).

Dataset Size	AES-256 (ms)	RSA-2048 (ms)	Blowfish (ms)
10 MB	50	120	65
100 MB	180	600	230
500 MB	750	2900	850

Table 4.1: Performance Comparison of Encryption Algorithms

AES-256 consistently outperformed RSA-2048 and Blowfish in terms of both encryption and decryption times, especially with larger datasets. RSA-2048 showed the highest computational overhead due to its asymmetric nature, making it less suitable for large-scale data encryption in real-time pipelines.

4.2 Security and Data Integrity Evaluation

To assess the effectiveness of the encryption algorithms, we evaluated the security strength and the ability of each algorithm to maintain data integrity during the transmission. The tests included man-in-the-middle attack simulations and decryption attempts with incorrect keys. The results of these tests are summarized in Table 4.2, which provides the percentage of successful data integrity checks and encryption resistance to attacks.

Algorithm	Data Integrity (%)	Attack Resistance (Score: 1-10)
AES-256	100	9.5
RSA-2048	100	9.0
Blowfish	99	8.5

Table 4.2: Security Strength and Integrity Evaluation

All three encryption algorithms maintained near-perfect data integrity, with AES-256 and RSA-2048 achieving 100% in tests. AES-256 showed the highest resilience to attack simulations, scoring 9.5 out of 10 in encryption resistance, indicating its robustness in securing sensitive data transmissions.

4.3 Resource Utilization and Scalability

The final analysis focused on the scalability of the encryption pipeline in terms of resource consumption, particularly memory and CPU utilization. Table 4.3 presents the average CPU and memory usage during encryption for different dataset sizes.

Dataset Size	AES-256 (CPU%)	RSA-2048 (CPU%)	Blowfish (CPU%)	AES-256 (Memory MB)	RSA-2048 (Memory MB)	Blowfish (Memory MB)
10 MB	10	25	15	50	60	55
100 MB	25	40	30	150	220	170
500 MB	50	75	55	400	600	450

Table 4.3: Resource Utilization for Encryption Pipelines

AES-256 demonstrated the most efficient use of CPU and memory resources, making it the most scalable solution for real-time encryption in data pipelines. RSA-2048, while highly secure, consumed significantly more CPU and memory resources, making it less viable for larger datasets in high-performance environments.

V. DISCUSSION

This paper has presented a comprehensive evaluation of three widely used encryption algorithms—AES-256, RSA-2048, and Blowfish—within the context of real-time, end-to-end encrypted data pipelines. The analysis considered performance metrics such as encryption and decryption times, CPU and memory utilization, and security strength, with the findings offering significant insights.

The results clearly demonstrate that **AES-256** outperforms RSA-2048 and Blowfish in terms of both computational efficiency and scalability. Across all dataset sizes, AES-256 showed faster encryption and decryption times, particularly excelling with larger datasets, where it encrypted a 500 MB file in 750 ms, compared to RSA-2048's 2900 ms. **RSA-2048**, while highly secure, exhibited the most significant computational overhead, making it less suitable for large-scale real-time data encryption. Blowfish performed moderately well, positioned between AES-256 and RSA-2048 in terms of speed, but showed higher resource consumption relative to AES-256.

In terms of security, **AES-256 and RSA-2048** both maintained 100% data integrity and demonstrated high resistance to attack simulations, with AES-256 achieving a slightly better resistance score of 9.5 out of 10. **Blowfish**, though nearly as secure, scored 99% in data integrity and 8.5 in attack resistance, suggesting it may be slightly more vulnerable in high-frequency data transmission environments.

Resource utilization tests revealed that **AES-256** consistently consumed the least amount of CPU and memory, making it the most efficient and scalable algorithm for high-throughput environments. In contrast, **RSA-2048** incurred significantly higher CPU usage (up to 75%) and memory consumption, which limits its scalability in resource-constrained systems. **Blowfish**

offered a middle-ground in terms of CPU and memory utilization but lagged behind AES-256 in overall efficiency.

5.2 Future Scope

While the study has provided valuable insights into the performance, security, and scalability of these encryption algorithms, there are several areas for future research to further enhance the understanding and applicability of end-to-end encryption in secure data pipelines.

1. **Algorithm Customization for Specific Use Cases**: Future research could explore the customization of encryption algorithms tailored to specific industries or application types. For instance, healthcare data might benefit from a different encryption strategy than financial data, based on varying regulatory requirements and risk tolerance levels.
2. **Hybrid Encryption Models**: The evaluation in this paper focused on standalone encryption algorithms, but combining symmetric and asymmetric encryption into hybrid models could offer an intriguing area of study. Future research could investigate hybrid encryption schemes that capitalize on the speed of AES-256 for bulk data encryption and the security advantages of RSA-2048 for key exchange.
3. **Quantum-Resistant Algorithms**: With the advancement of quantum computing, traditional encryption algorithms like RSA may become vulnerable. Future research should explore post-quantum encryption methods and their integration into real-time data pipelines.
4. **Optimizing for Cloud and Edge Computing**: As cloud and edge computing continue to expand, the encryption needs for distributed systems are growing. Future work could evaluate how encryption algorithms perform in edge computing environments where computational resources are limited and real-time encryption is crucial.

By addressing these areas, future research can further optimize end-to-end encryption for a broader range of applications and ensure that encryption techniques continue to evolve alongside emerging technologies.

VI. CONCLUSION

This study provides a comprehensive analysis of three encryption algorithms—AES-256, RSA-2048, and Blowfish—within an end-to-end encrypted data pipeline, evaluating their suitability for handling sensitive data in real-time, high-throughput environments. AES-256 emerged as the optimal choice, consistently outperforming RSA-2048 and Blowfish across various metrics. It demonstrated faster encryption times, completing the task in 750 ms for a 500 MB dataset, while RSA-2048 required 2900 ms and Blowfish needed 850 ms. Additionally, AES-256 proved to be the most resource-efficient, utilizing 50% of CPU resources for large datasets, compared to RSA-2048's 75%. In terms of security, AES-256 maintained 100% data integrity and achieved a 9.5 out

of 10 in attack resistance tests. While RSA-2048 offers strong security, its high computational cost makes it less suitable for large-scale data encryption. Blowfish, though relatively efficient, demonstrated slightly weaker performance in both speed and security. Based on these results, AES-256 is recommended for industries requiring secure and scalable data pipelines, offering an optimal balance between encryption strength, resource efficiency, and real-time performance.

REFERENCES

[1] Zhu, Jinwei, et al. "Full Encryption: An end to end encryption mechanism in GaussDB." *Proceedings of the VLDB Endowment* 14.12 (2021): 2811-2814.

[2] Burkhalter, Lukas, et al. "Zeph: Cryptographic enforcement of end-to-end data privacy." *15th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 21)*. 2021.

[3] Jan, Mian Ahmad, et al. "SmartEdge: An end-to-end encryption framework for an edge-enabled smart city application." *Journal of Network and Computer Applications* 137 (2019): 1-10.

[4] Schillinger, Fabian, and Christian Schindelhauer. "End-to-end encryption schemes for online social networks." *Security, Privacy, and Anonymity in Computation, Communication, and Storage: 12th International Conference, SpaCCS 2019, Atlanta, GA, USA, July 14–17, 2019, Proceedings 12*. Springer International Publishing, 2019.

[5] Kumar, Sam, et al. "{JEDI}:{Many-to-Many}{End-to-End} encryption and key delegation for {IoT}." *28th USENIX security symposium (USENIX Security 19)*. 2019.

[6] Vanin, Fausto Neri da Silva, et al. "A blockchain-based end-to-end data protection model for personal health records sharing: a fully homomorphic encryption approach." *Sensors* 23.1 (2022): 14.

[7] Basem, Omar, Abrar Ullah, and Hani Ragab Hassen. "Stick: an end-to-end encryption protocol tailored for social network platforms." *IEEE Transactions on Dependable and Secure Computing* 20.2 (2022): 1258-1269.

[8] Reuter, Adrian, et al. "Usability of end-to-end encryption in e-mail communication." *Frontiers in big Data* 4 (2021): 568284.

[9] Dechand, Sergej, et al. "In encryption we don't trust: The effect of end-to-end encryption to the masses on user perception." *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2019.

[10] Kamara, Seny, et al. "Outside looking in: Approaches to content moderation in end-to-end encrypted systems." *arXiv preprint arXiv:2202.04617* (2022).

[11] Schwenk, Jörg, et al. "Mitigation of attacks on email end-to-end encryption." *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 2020.

[12] Wei, Jianghong, et al. "Enabling (end-to-end) encrypted cloud emails with practical forward secrecy." *IEEE Transactions on Dependable and Secure Computing* 19.4 (2021): 2318-2332.

[13] Hale, Britta, and Chelsea Komlo. "On end-to-end encryption." *Cryptology ePrint Archive* (2022).

[14] Hassani Karbasi, Amir, and Siyamak Shahpasand. "SINGLETON: A lightweight and secure end-to-end encryption protocol for the sensor networks in the Internet of Things based on cryptographic ratchets." *The Journal of Supercomputing* 77.4 (2021): 3516-3554.

[15] Singh, Raman, Ark Nandan Singh Chauhan, and Hitesh Tewari. "Blockchain-enabled end-to-end encryption for instant messaging applications." *2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, 2022.