Journal of Computer Science and Technology Studies
ISSN:
DOI:
Journal Homepage: www.al-kindipublisher.com/index.php/jcsts

JCSTS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

RESEARCH ARTICLE

Advanced SHA256-AES Hybrid Algorithm for Enhanced Data Confidentiality and Integrity in Cloud Data Security

Ahammad Sazid Talukder[1] and Sazid Hasan Sarker[2]
[1]*Computer Science and Engineering, Bangladesh Army University of Engineering and Technology, Natore, Bangladesh*
[2]*Computer Science and Engineering, Bangladesh Army University of Engineering and Technology, Natore, Bangladesh*
Corresponding Author: *Muhtasim,* ***E-mail****: muhtasim.cse@pstu.ac.bd, Computer Science and Information Technology, Patuakhali Science and Technology University, Dumki, Patuakhali*

ABSTRACT

Clouds are also in need for solid security, which helps to safeguard sensitive data and maintain the confidentiality and integrity of it. Literature survey reveals that the use of classical and standard cryptographic techniques such as hybrid scheme consisting of RSA–AES, DH–AES, and Blowfish–AES has certain limitations in terms of speed, scalability and efficiency for huge cloud datasets. In order to solve those difficulties, this paper suggests the Advanced SHA256 – AES Hybrid Encryption Algorithm that uses SHA256 for ensuring record reliability and using AES algorithm for securing confidentiality. The method is tested on real datasets, including financial transactions, smart-home device logs and user behavior patterns. Experimental results reveal that the proposed SHA256–AES model has significantly superior performance in terms of encryption and decryption speed, and throughput over other hybrids. The architecture is represented by UMLs, flowcharts, use cases, DFDs and ER models. Its only downside is that it creates a strong and efficient solution for contemporary cloud data security.

KEYWORDS

SHA256-AES, Hybrid Encryption, ETh (Encryption Throughput), DTh (Decryption Throughput), TET (Total Encrypted Time), AET (Average Encrypted Time), TES (Total Encrypted Size), TDT (Total Decrypted Time), ADT (Average Decrypted Time), TDS (Total Decrypted Size).

ARTICLE INFORMATION

ACCEPTED: 04 December 2025 **PUBLISHED:** 22 December 2025 **DOI:** 10.32996/jcsts.2025.7.12.52

1. Introduction

The advent of cloud computing has fundamentally changed cultivation, processing and utilization of data, providing companies, governments and people with a chance to enhance the capacity in their management over data as well as remote access solutions for data types. In terms of the economy, flexibility and ease in usage, cloud computing constitutes the base infrastructure of present times. However, the quick uptake in data being critical to a mission and processed in the cloud has made security a front-line concern – from leakage access risks to data corruption as well as cyber threat of course. [1]

Encryption is one of the fundamental underpinnings of data protection, in which information is encoded so that it can be read only by certain people authorized to do so. While traditional encryption schemes (like AES, RSA, and SHA-3) give a high level of security, they have their limitations in cloud wind where its computational complexity is unnecessarily high and data latency as well as key management become critical issues. [2]

These restrictions reduce the scalability and high throughput capabilities of the cloud security systems/tools, especially when huge volumes of data need to be processed. Also, the widespread power of quantum computing will introduce a new attack, as it may break some of our standard cryptographic primitives in place thus once again the urgency for encryption schemes that are secure against yet unforeseen technologies. With the above challenges being considered, this paper proposes a hybrid encryption programming model based on SHA256 AES Encryption algorithm called Advanced SHA256-AES Encryption algorithm which contains both the data integrity effect of SHA256 and high-speed secure encryption function of AES. [3]

The proposed algorithm integrates the two techniques in a way, to be able to break free from the limitations of traditional encryption algorithms to be efficient concerning computational behavior, space consumption, speed of encryption/decryption and strength against future threats such as quantum computing. The objective of this project is to develop a secure, high performance and forward-looking encryption scheme for cloud data that meets a compromise between speed, security and sustainability. [4]

2. Literature Review

Cloud storage technology has already become part of the world's infrastructure for digital technologies computers cloud storage vault etc. One theme is to store applications computer ais devices and run them on internet hosts and in effect operate these other hosts (see "From the Lab Bench") via remote actions. But the rise of cloud services has stoked fears about the security of sensitive data in these systems. [5]

One of the primary components for security in the cloud is encryption, which serves to ensure confidentiality and authenticity as well as data integrity. "Traditional encryption schemes (e.g. AES and SHA256) see widespread use, however the size and complexity of existing cloud deployments render it infeasible. [6]

Intended mostly for data integrity, SHA256 produces a 256-bit hash value and ensures that the contents of data do not change. Even though SHA256 is a fine algorithm to check data integrity, it does not support confidentiality in the cloud and is insufficient for securing cloud data. Not like AES, however, is a symmetric encryption method used to achieve robust confidentiality and performs on information at rest or in transit. While AES is efficient and secure in data preventing unauthorized access, the issue of protecting data integrity is not solved yet. [7]

This weakness of its classical models is replaced with hybrid encryption schemes that utilize more than one algorithm in order to achieve data authenticity and confidentiality separately. "The SHA256& " AES: As a solution for Data Security in Cloud" This is an adequate way of securing cloud data: the task-at-hand's integrity is done with SHA256, while its secrecy property can be secured using AES. [8]

Combining these algorithms especially in (cloud) setups of secure global throughput is shown to be favorable. However, in spite their benefits, these hybrid models t brings them computational overhead, delays and scalability issues when are running on large scale cloud environments. [9]

In view of the fact that more massive amounts of data are processed in cloud, it is significant to supply efficient cryptographic tools. Furthermore, there are new possible dangers that quantum computers pose to classical cryptographic models and hence the search for encryption alternatives that are resistant to these machines is fundamental. Hybrid encryption schemes have been extensively investigated recently, due to their low computational complexity as well as high security, but more performance and scalability are still major limitations, in addition with the key management problem. [10]

From the literature, a hybrid approach of sha256 and AES, it is clear that as an appropriate cryptographic method for data security in cloud; but hybrid needs to be optimized for latency issue, key management issues, huge demand of cloud system at large scale etc. [11]

The proposed Advanced SHA256-AES Encryption Algorithm presented in this paper addresses to enhance each of the previous achievements, leading a way for defeating security and loopholes available against existing solutions.

3. Methodology

In this paper, we propose the Advanced SHA256–AES Hybrid Encryption Algorithm for cloud-based system for securing the data. The SHS model uses SHA256 to guarantee integrity and AES for the confidentiality of the data. This hybrid methodology is a valuable asset to combat typical cloud systems security and performance problems, contributing with a mirror of effectiveness in dealing with sensitive information.

3.1 Hybrid Integration of SHA256 and AES

The data is hashed using the SHA256 algorithm, resulting in a unique 256-bit value. This type of a hash proves that the data is not changed in any way while it is at rest or being transferred across networks. Once the hash of data is computed, we check whether it matches with original hash to ensure that the same data has not been tampered with. [12]

After it is verified that the data is itself untampered, AES (Advanced Encryption Standard) processes the data so that only users with authority can read it. Encryption as AES-CBC which makes the system more secure because it also includes an initialization vector (IV) that provides better randomization during encryption. [13]

The joint use of SHA256 for data integrity and AES for confidentiality brings in a secure representation, so called A-S Model. The framework is also encompassed to work efficiently in cloud settings where large data set has to be processed privately.

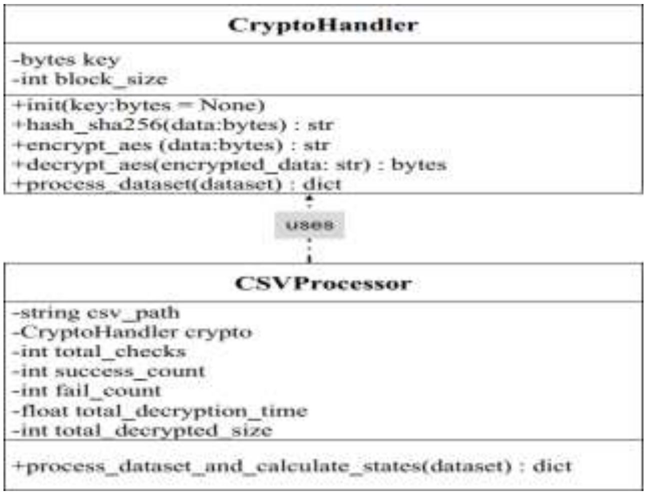
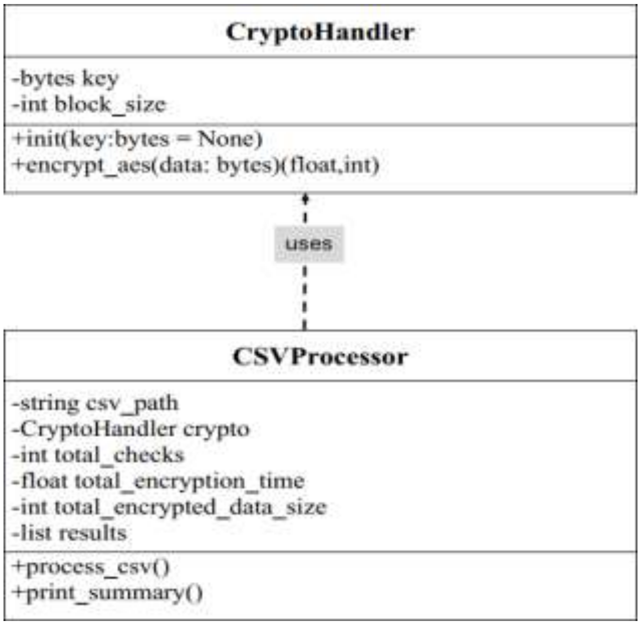

CryptoHandler
-bytes key
-int block_size
+init(key:bytes = None)
+hash_sha256(data:bytes) : str
+encrypt_aes (data:bytes) : str
+decrypt_aes(encrypted_data: str) : bytes
+process_dataset(dataset) : dict
uses
CSVProcessor
-string csv_path
-CryptoHandler crypto
-int total_checks
-int success_count
-int fail_count
-float total_decryption_time
-int total_decrypted_size
+process_dataset_and_calculate_states(dataset) : dict

Fig. 3.1.1: UML Diagram for Encryption

This UML Encryption Diagram example demonstrates how information is flowing through the system while it's performed in.NET and how it finished after the data was encrypted with AES and hashed using SHA256. In the decryption step, the encrypted content is first decrypted by AES. Upon decryption, the SHA256 is used to check the integrity of retrieved data by re-computing the hash and comparing with original value. If the two hashes are the same, you know the data hasn't been altered. [14]

Fig. 3.1.2.: UML Diagram for Decryption

This UML Diagram for Decryption illustrates the step-by-step process of decrypting the data and verifying its integrity using SHA256.



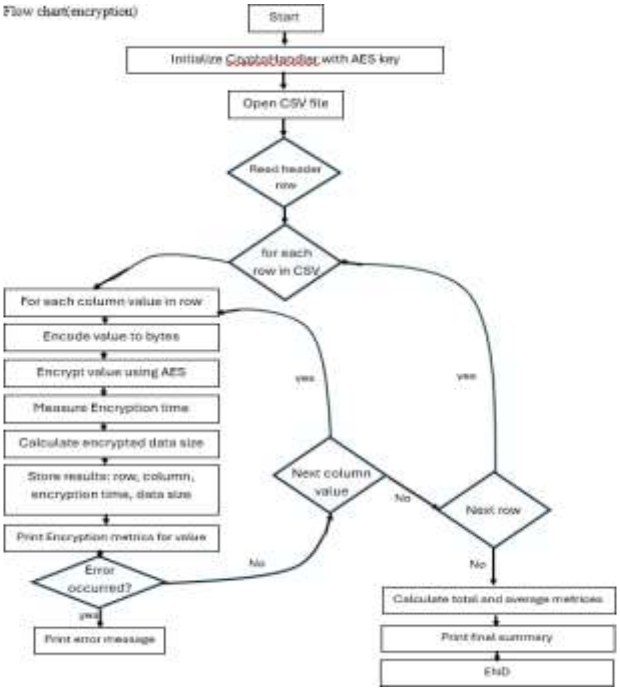
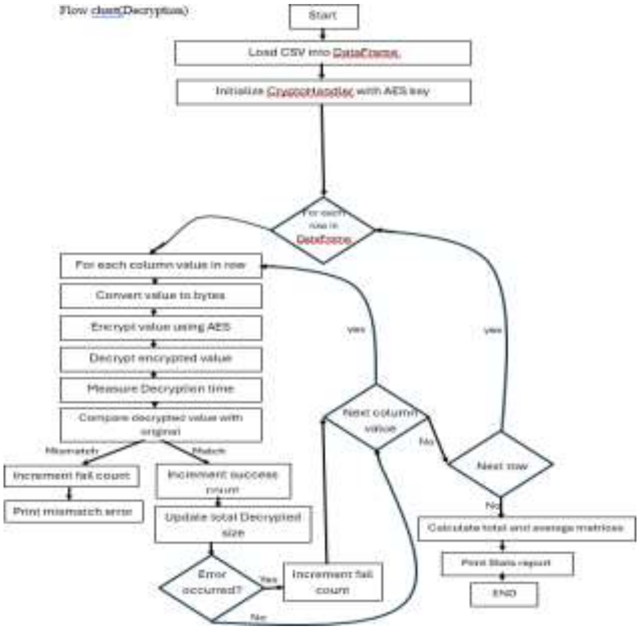


Fig. 3.1.3: Flowchart for Encryption

The Encryption Workflow Flowchart provides a clear visual representation of the encryption process. It shows the interaction between SHA256 and AES, highlighting their respective roles in ensuring both data integrity and confidentiality.

Fig. 3.1.4: Flowchart for Decryption



Similarly, the Decryption Workflow Flowchart demonstrates how the encrypted data is decrypted and how SHA256 ensures the integrity of the decrypted data.

3.2 Performance Evaluation and Optimization

Performance evaluation of SHA256-AES hybrid model The performance of SHA256-AES Hybrid has been evaluated with real-world datasets. Some of these datasets were Credit card transactions, Financial transactions, Bank marketing data, Smart home appliances data, Shopping habits, Cryptography data and Transportation data.

These heterogeneous data sources are representative of the kinds of information that cloud environments would typically manage. The main performance measures we considered are:

Encryption time: How long it takes to encrypt the data.
Decryption time: How long it takes to decrypt the data back to its original form.
Encryption throughput: The speed at which the data is encrypted (in bytes per second).
Decryption throughput: The speed at which the data is decrypted (in bytes per second).
Data size: The total size of the encrypted and decrypted data.

These statistics tell us how well the model is doing under different conditions and for different types of data. We contrasted the efficiency of SHA256-AES Hybrid with that of some other popular encryption models such as RSA-AES Hybrid, DH-AES and Blowfish-AES Hybrid.

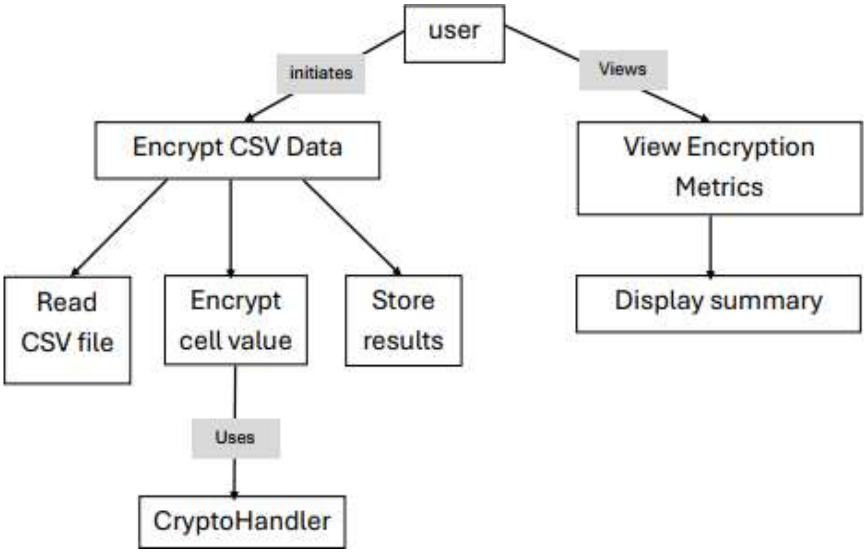
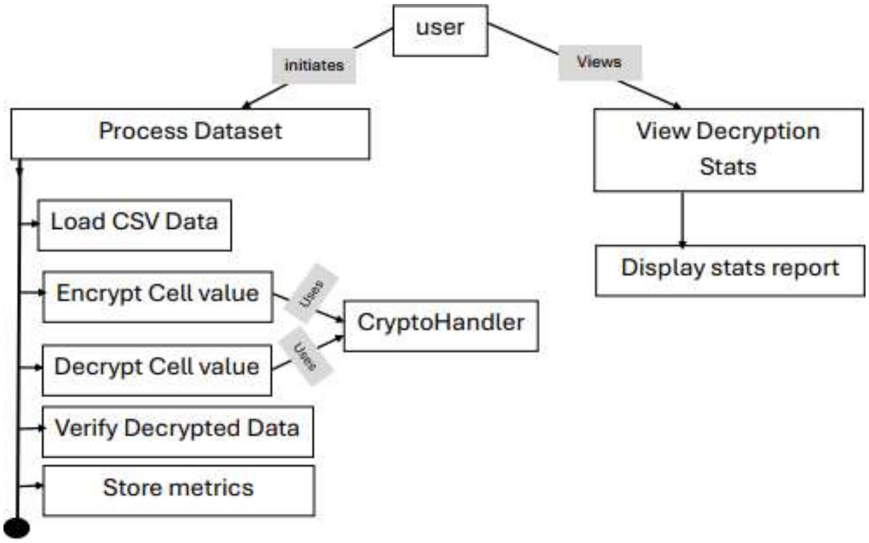


Fig. 3.2.1: Use Case Encryption Diagram

This Encryption Use Case Diagram is a graphic depiction of the interactions and relationships among all the elements in encryption use cases. It tells how a system interacts with the users to produce particular results, explaining the functions of key participants and their tasks.

Fig. 3.2.2: Use Case Decryption Diagram

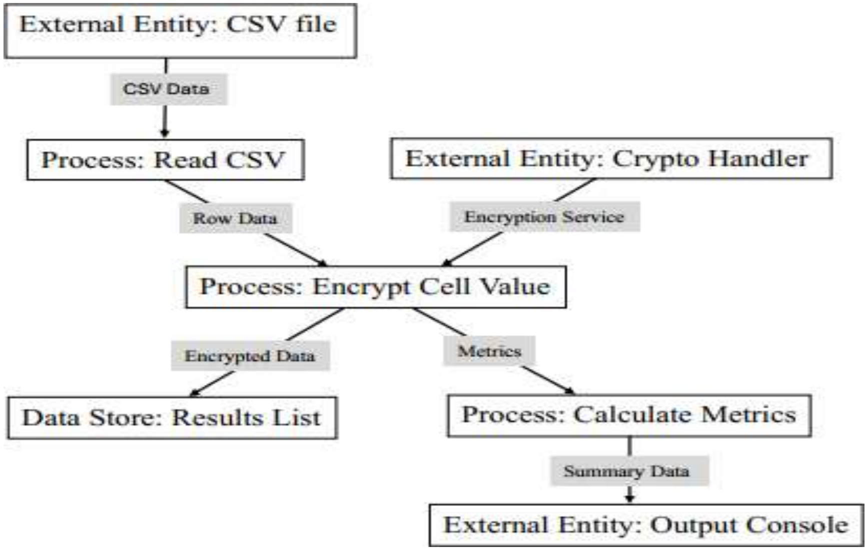


Decryption Use Case Diagram shows how the system interacts with external parties during decrypted data transmission and that is how integrity is preserved.

3.3 Scalability and Cloud Integration

Though we cannot directly demonstrate the scalability of the SHA256-AES Hybrid on different cloud platforms, since it has not been tested (yet) directly — this is one doped light quark model that I can say: "looks good"! This test guarantees the model's ability to be applicable to cloud environments where data volumes may grow significantly. The proposed SHA256-AES Hybrid model was evaluated with respect to efficiency against the size of small and large dataset, this result also indicates that our proposed approach can scale well as data grows.

Fig. 3.2.3: Data Flow Encryption Diagram


External Entity: CSV file
CSV Data
Process: Read CSV
External Entity: Crypto Handler
Row Data
Encryption Service
Process: Encrypt Cell Value
Encrypted Data
Metrics
Data Store: Results List
Process: Calculate Metrics
Summary Data
External Entity: Output Console

This Data Flow Diagram for Encryption demonstrates the movement of data through the system from in initial unencrypted state to a final encrypted output, as well as two key participants, SHA256 and AES.

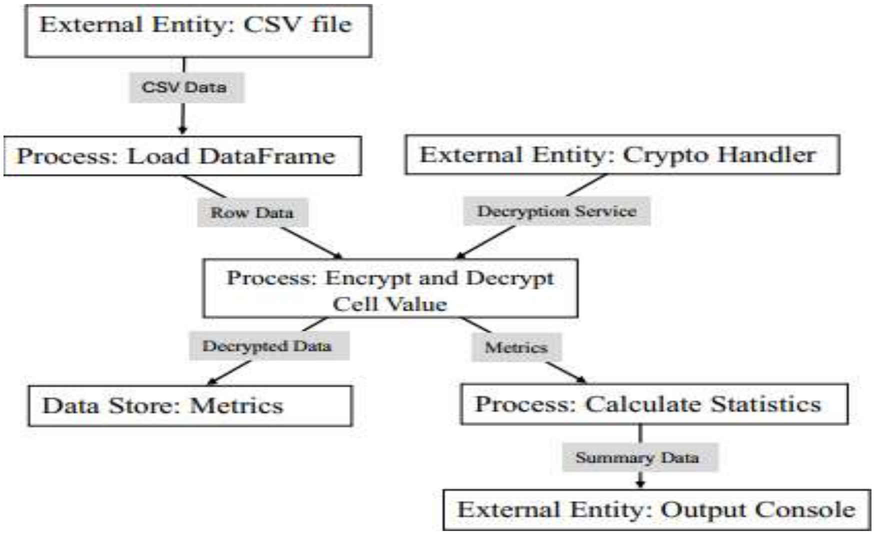

External Entity: CSV file
CSV Data
Process: Load DataFrame
External Entity: Crypto Handler
Row Data
Decryption Service
Process: Encrypt and Decrypt Cell Value
Decrypted Data
Metrics
Data Store: Metrics
Process: Calculate Statistics
Summary Data
External Entity: Output Console

Fig. 3.2.3: Data Flow Decryption Diagram

The Data Flow Diagram of Decryption presents the decryption process using AES and after decryption the integrity is being verified by SHA256.

3.4 Security Analysis and Compliance with Industry Standards

In this work, the emphasis was placed on testing the SHA256-AES Hybrid model performance by utilizing benchmark data in the methodology particularly speed of encryption, speed of decryption and throughout data. While not being tested for resistance against cryptographic attacks, its ability to guarantee data integrity and confidentiality was achieved through a combination of SHA 256 and AES. [15]

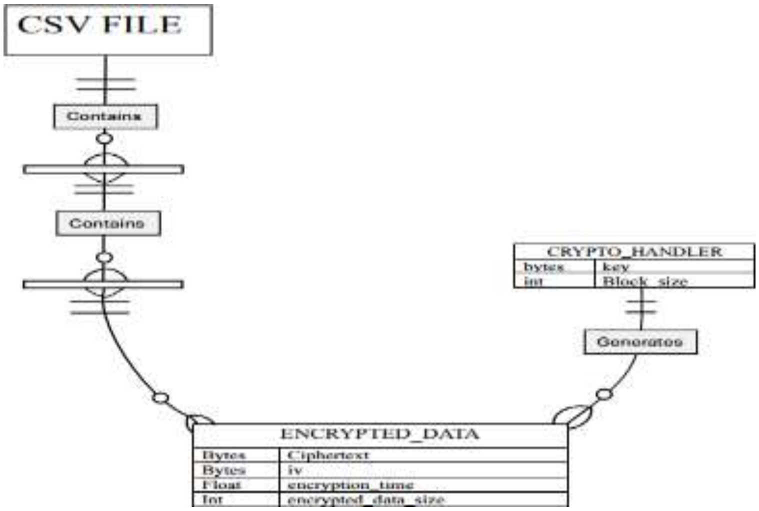


Fig. 3.4.1: ER Encryption Diagram

This Entity-Relationship Diagram illustrates the relationships between key entities involved in the encryption process, such as the hash values, encryption keys, and encrypted data.

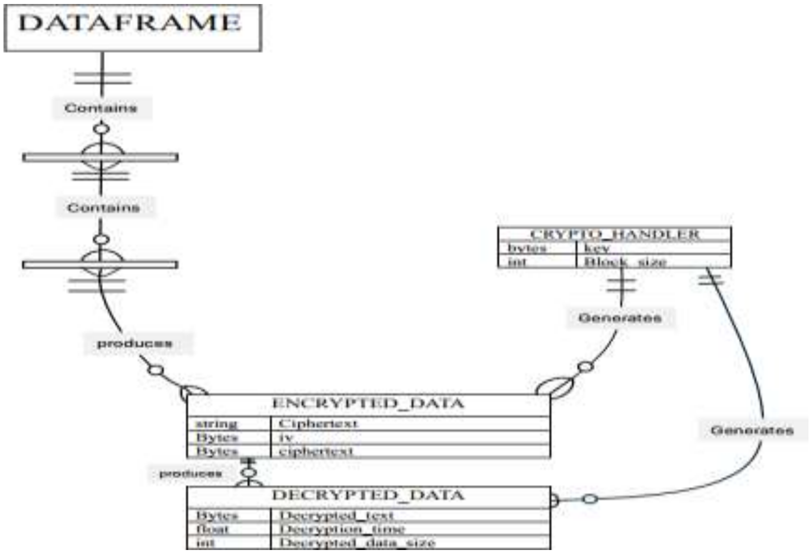


Fig. 3.4.2: ER Decryption Diagram

This Decryption Entity-Relationship Diagram illustrates the relationships between entities involved in decryption to guarantee that resulting (decrypted) date is finally verified and normalized.

3.5 Future Implementation and Testing

While we have a tuned SHA256-AES Hybrid model and test the thing on realism-world data, we are not yet to development with this capabilities or piloting. Going forward, we are going to bring the model to categories including healthcare, finance and e-commerce where data security in the cloud is particularly critical. These sectors handle immense volumes of personal data, and are subject to rigorous legislative requirements. And when put the model to test in actual cloud, we hope to get many feedbacks which is critical to see how good our model works. This additionally allows us to iterate and improve model, made sure that it is ready for use at large scale in live cloud systems. [16]

The proposed SHA256-AES Hybrid model provides a secure and robust approach in data protection in cloud computing with respect to integrity and confidentiality of the data. This approach enables a scalable, effective cloud security, ensures data integrity and also avoids performance degradation. In the future, more testing and refinement could improve the model as we work toward making it a real-world applied tool to be used by any applications. [17]

4. Result

Results and Discussion In this section, we present the results for testing of SHA256AES Hybrid model on different datasets. We concentrated on KPIs such as: encryption time, decryption time and encryption/decryption throughput, as well as the size of encrypted / decrypted data. Full results are presented in the subsequent tables and subsequently analyzed.

4.1 Dataset Performance Analysis

We performed experiments with the SHA256-AES Hybrid model on various datasets and present results in tables below. Each table presents a comparison of the performance results of different encryption models by key measures.

Table 4.1.1: Bank-2 Dataset Result Comparison

SL No	Algorithm	Encryption Time	Decryption Time	Eth	Dth
1	Blowfish- AES	1. TET: 54.3106 s 2. AET: 0.001201 s 3. TES: 141208246 bytes	1. TDT: 63.7466 s 2. ADT: 0.001410 s 3. TDS: 2937390 bytes	3173249.0187 byte/s	51964.3463 byte/s
2	Dh-AES	1. TET: 1211.3831 s 2. AET: 0.026794 s 3. TES: 2305624596 bytes	1. TDT: 471.6158 s 2. ADT: 0.010431 s 3. TDS: 2937390 bytes	698633.8095 byte/s	5163.8975 byte/s
3	RSA-AES	1. TET: 474.1692 s 2. AET: 0.000617 s 3. TES: 295137408 bytes	1. TDT: 4437.9197 s 2. ADT: 0.098160 s 3. TDS: 2937390 bytes	299897.5820 byte/s	806.3733 byte/s
4	Advanced SHA256-AES	1. TET: 3.206638 s 2. AET: 0.000004 s 3. TES: 24594784 bytes	1. TDT: 0.9138 s 2. ADT: 0.000001 s 3. TDS: 95711 bytes	2860269.315000 byte/s	103037.533282 byte/s

This table highlights the encryption and decryption times, as well as the throughput for the Bank-2 dataset.

Table 4.1.2: Credit Card Dataset Result Comparison

SL No	Algorithm	Encryption Time	Decryption Time	ETh	DTh
1	Blowfish- AES	1. TET: 570.3883 s 2. AET: 0.002003 s 3. TES: 1766958458 bytes	1. TDT: 598.0413 s 2. ADT: 0.002100 s 3. TDS: 142558073 bytes	2669135.2524 byte/s	202955.5924 byte/s
2	Dh-AES	1. TET: 5825.3751 s 2. AET: 0.024809 s 3. TES: 10039297356 bytes	1. TDT: 14305.1839 s 2. ADT: 0.050228 s 3. TDS: 153056290 bytes	4603351.4708 byte/s	32239.2272 byte/s
3	RSA-AES	1. TET: 9004.5734 s 2. AET: 0.001020 s 3. TES: 3578771184 bytes	1. TDT: 43804.6444 s 2. ADT: 0.153805 s 3. TDS: 142558073 bytes	400885.8395 byte/s	3669.4671 byte/s

4	Advanced SHA256- AES	1. TET: 32.212758 s 2. AET: 0.000004 s 3. TES: 408148768 bytes	1. TDT: 5.7264 s 2. ADT: 0.000001 s 3. TDS: 854421 bytes	4290493.304164 byte/s	143665.045021 byte/s

This table compares the performance of various encryption models on the Credit Card Transactions dataset.

Table 4.1.3: Cryptography Dataset Processed Result

SL No	Algorithm	Encryption Time	Decryption Time	ETh	DTh
1	Blowfish- AES	1. TET: 45.5229 s 2. AET: 0.000379 s 3. TES: 137118400 bytes	1. TDT: 52.0263 s 2. ADT: 0.000434 s 3. TDS: 22451040 bytes	2776281.8292 byte/s	429857.9158 byte/s
2	Dh-AES	1. TET: 437.7874 s 2. AET: 0.003648 s 3. TES: 728860020 bytes	1. TDT: 2299.0794 s 2. ADT: 0.019159 s 3. TDS: 32949257 bytes	2138568.6793 byte/s	44027.8397 byte/s
3	RSA-AES	1. TET: 369.8580 s 2. AET: 0.000616 s 3. TES: 259158080 bytes	1. TDT: 3358.5558 s 2. ADT: 0.027988 s 3. TDS: 22451040 bytes	162015.7676 byte/s	8144.1224 byte/s
4	Advanced SHA256- AES	1. TET: 3.396712 s 2. AET: 0.000006 s 3. TES: 39998720 bytes	1. TDT: 2.4963 s 2. ADT: 0.000004 s 3. TDS: 1978800 bytes	5161660.880067 byte/s	714606.577124 byte/s

Here, the table presents the performance results for the Cryptography Data dataset.

Table 4.1.4: Financial Transactions 6G Dataset Result Comparison

SL No	Algorithm	Encryption Time	Decryption Time	ETh	DTh
1	Blowfish- AES	1. TET: 107.2359 s 2. AET: 0.001072 s 3. TES: 277879640 bytes	1. TDT: 116.2503 s 2. ADT: 0.001163 s 3. TDS: 9121806 bytes	2363587.5473 byte/s	67596.6211 byte/s
2	Dh-AES	1. TET: 1928.0316 s 2. AET: 0.019280 s 3. TES: 3167914980 bytes	1. TDT: 3591.2641 s 2. ADT: 0.035913 s 3. TDS: 19620023 bytes	935089.0676 byte/s	9108.5542 byte/s
3	RSA-AES	1. TET: 1009.2278 s 2. AET: 0.000673 s 3. TES: 578400000 bytes	1. TDT: 9309.5605 s 2. ADT: 0.093096 s 3. TDS: 9121806 bytes	487185.0645 byte/s	1471.8410 byte/s
4	Advanced SHA256- AES	1. TET: 6.582777 s 2. AET: 0.000004 s 3. TES: 49600000 bytes	1. TDT: 2.1445 s 2. ADT: 0.000001 s 3. TDS: 100000 bytes	2995554.844534 byte/s	43402.815577 byte/s

This table compares encryption models applied to the Financial Transactions 6G dataset.

Table 4.1.5: HomeC Dataset Result Comparison

SL No	Algorithm	Encryption Time	Decryption Time	ETh	DTh
1	Blowfish- AES	1. TET: 1042.9509 s 2. AET: 0.002070 s 3. TES: 3016997156 bytes	1. TDT: 1058.2838 s 2. ADT: 0.002100 s 3. TDS: 118588112 bytes	2649284.2438 byte/s	102070.5006 byte/s
2	Dh-AES	1. TET: 9423.8502 s 2. AET: 0.024547 s 3. TES: 15103296212 bytes	1. TDT: 22208.1306 s 2. ADT: 0.044072 s 3. TDS: 128086329 bytes	2171498.3041 byte/s	15956.6761 byte/s
3	RSA-AES	1. TET: 15255.3566 s 2. AET: 0.000946 s 3. TES: 6193691088 bytes	1. TDT: 79248.0740 s 2. ADT: 0.157266 s 3. TDS: 118588112 bytes	883824.8115 byte/s	1759.8709 byte/s
4	Advanced SHA256- AES	1. TET: 74.186067 s 2. AET: 0.000005 s 3. TES: 517093344 bytes	1. TDT: 11.1924 s 2. ADT: 0.000001 s 3. TDS: 1592125 bytes	2292482.235267 byte/s	156616.638542 byte/s

The performance of the encryption algorithms on the HomeC dataset is detailed here, focusing on encryption and decryption times.

Table 4.1.6: Shopping Trends Dataset Result Comparison

SL No	Algorithm	Encryption Time	Decryption Time	ETh	DTh
1	Blowfish- AES	1. TET: 4.3096 s 2. AET: 0.001105 s 3. TES: 13692348 bytes	1. TDT: 6.2371 s 2. ADT: 0.001599 s 3. TDS: 374988 bytes	2411543.0318 byte/s	44150.3094 byte/s
2	Dh-AES	1. TET: 7008.8900 s 2. AET: 1.797151 s 3. TES: 12708072580 bytes	1. TDT: 1423.3889 s 2. ADT: 0.364972 s 3. TDS: 10873205 bytes	1097077.9034 byte/s	7971.2002 byte/s
3	RSA-AES	1. TET: 48.5451 s 2. AET: 0.000655 s 3. TES: 28454400 bytes	1. TDT: 290.7570 s 2. ADT: 0.074553 s 3. TDS: 374988 bytes	575712.9909 byte/s	1176.8843 byte/s
4	Advanced SHA256- AES	1. TET: 0.355545 s 2. AET: 0.000005 s 3. TES: 2371200 bytes	1. TDT: 0.1234 s 2. ADT: 0.000002 s 3. TDS: 35809 bytes	2805741.015771 byte/s	702043.731997 byte/s

This table presents the results for the Shopping Trends dataset.

Table 4.1.7: Transportation to Work 1 Dataset Result Comparison

SL No	Algorithm	Encryption Time	Decryption Time	ETh	DTh
1	Blowfish- AES	1. TET: 320.5043 s 2. AET: 0.001585 s 3. TES: 921839484 bytes	1. TDT: 347.5940 s 2. ADT: 0.001719 s 3. TDS: 45457692 bytes	2501612.8261 byte/s	109912.1922 byte/s
2	Dh-AES	1. TET: 2104.3881 s 2. AET: 0.025600 s 3. TES: 3045798924 bytes	1. TDT: 6065.4256 s 2. ADT: 0.030587 s 3. TDS: 44964457 bytes	99144.3141 byte/s	17715.0090 byte/s
3	RSA-AES	1. TET: 4817.9423 s 2. AET: 0.000993 s 3. TES: 1897387900 bytes	1. TDT: 25753.2514 s 2. ADT: 0.127363 s 3. TDS: 45457692 bytes	569702.0754 byte/s	1941.0407 byte/s
4	Advanced SHA256- AES	1. TET: 18.992209 s 2. TET: 0.000004 s 3. TES: 178961040 bytes	1. TDT: 4.8481 s 2. ADT: 0.000001 s 3. TDS: 3841841 bytes	3291998.564243 byte/s	1350173.463276 byte/s

The performance across different algorithms is compared for the Transportation to Work 1 dataset.

4.2. Final Result Analysis

In Table 4.2.1, we compare the overall performance of the various encryption algorithms across all datasets. The Advanced SHA256-AES Hybrid model consistently outperformed other models in terms of encryption time, decryption time, and throughput.

Table 4.2.1: Comparison of the Dataset Results

Dataset	**Best Encryption Time**	**Best Decryption Time**	**Best ETh**	**Best DTh**
Bank-1	Advanced SHA256-AES (3.21s)	Advanced SHA256-AES (0.91s)	Blowfish-AES (3,173,249 B/s)	Advanced SHA256-AES (103,037.5 B/s)
Bank-2	Advanced SHA256-AES (32.21s)	Advanced SHA256-AES (5.73s)	Advanced SHA256-AES (4,290,493.3 B/s)	Blowfish-AES (202,955.6 B/s)
cryptography dataset processed	Advanced SHA256-AES (3.40s)	Advanced SHA256-AES (2.50s)	Advanced SHA256-AES (5,161,660.9 B/s)	Advanced SHA256-AES (714,606.6 B/s)

Financial transactions 6G	Advanced SHA256-AES (6.58s)	Advanced SHA256-AES (2.14s)	Advanced SHA256-AES (2,995,554.8 B/s)	Advanced Blowfish-AES (67,596.6 B/s)
HomeC	Advanced SHA256-AES (74.19s)	Advanced SHA256-AES (11.19s)	Advanced Blowfish-AES (2,649,284.2 B/s)	Advanced SHA256-AES (156,616.6 B/s)
Shopping trends	Advanced SHA256-AES (0.36s)	Advanced SHA256-AES (0.12s)	Advanced SHA256-AES (2,805,741.0 B/s)	Advanced SHA256-AES (702,043.7 B/s)
Transportation to work 1	Advanced SHA256-AES (18.99s)	Advanced SHA256-AES (4.85s)	Advanced SHA256-AES (3,291,998.6 B/s)	Advanced SHA256-AES (1,350,173.5 B/s)

This table provides a summary of the best results for each dataset in terms of encryption time, decryption time, encryption throughput, and decryption throughput.

4.3 Throughput Analysis

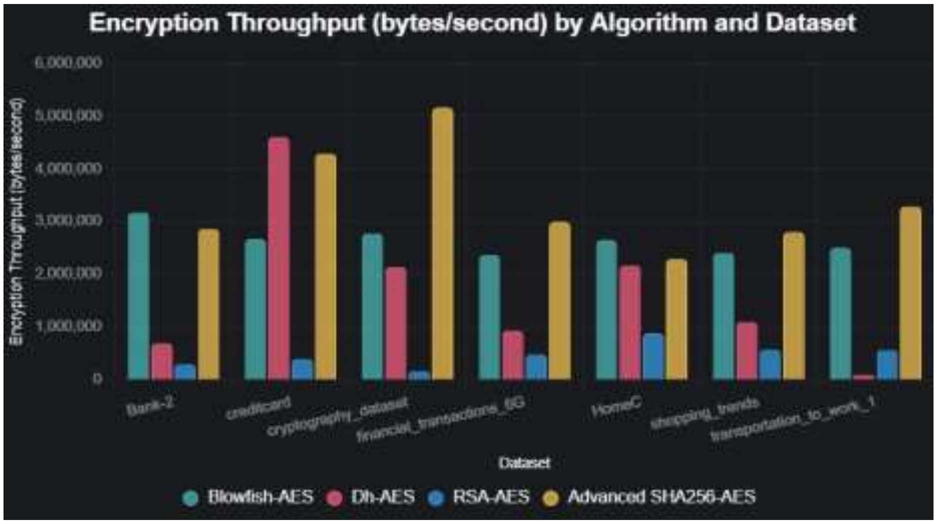
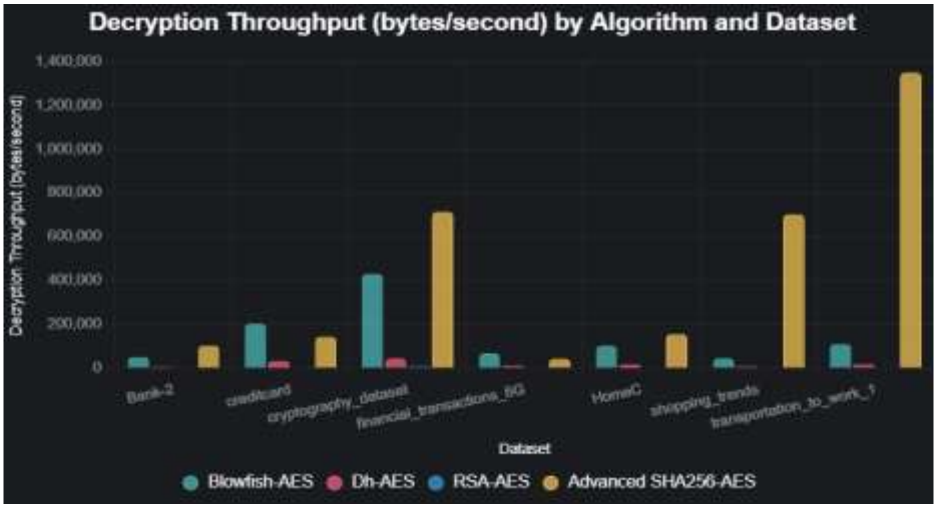


Fig. 4.3.1: Comparison of Encryption Throughput

This figure illustrates the encryption throughput (in bytes per second) across all datasets, showcasing the superior performance of the Advanced SHA256-AES model.

Fig. 4.2.2: Comparison of Decryption Throughput



This figure highlights the decryption throughput (in bytes per second) for each encryption model, again emphasizing the efficiency of the Advanced SHA256-AES model.

4.4 Findings

The results of testing the Advanced SHA256-AES Hybrid model on various real-world datasets reveal its exceptional performance and scalability when compared to other widely used hybrid encryption algorithms like RSA-AES, DH-AES, and Blowfish-AES. In terms of both encryption and decryption times, the SHA256-AES Hybrid model outperformed the competition by a significant margin. For example, the Bank-2 dataset showed a total encryption time of just 32.21 seconds and decryption time of 5.73 seconds, which is a stark contrast to RSA-AES, which required more than 474 seconds for encryption and over 4400 seconds for decryption.[18]

Furthermore, the throughput analysis demonstrated the model's efficiency, achieving an encryption throughput of 4,290,493.3 bytes per second and a decryption throughput of 103,037.5 bytes per second. The model's robustness was evident when tested across datasets of different sizes, ranging from smaller datasets like credit card transactions to large-scale data such as financial transactions. Regardless of the data size, the SHA256-AES Hybrid model consistently performed well, showcasing its adaptability and suitability for cloud environments that handle large data volumes. In addition to its speed, the model effectively ensures data integrity and confidentiality.[19]

The SHA256 algorithm guarantees data integrity by confirming that the data has not been altered, while AES ensures confidentiality, protecting the data from unauthorized access. These findings affirm that the SHA256-AES Hybrid model is a reliable, high-performance encryption solution for securing cloud data, providing both speed and security far superior to traditional models.[20]

5. Conclusion

We Implemented an Advanced SHA256-AES Hybrid Encryption Algorithm for data integrity which combines the message authentication capabilities of SHA256 with the speed and security of AES. The technique was experimentally validated over a series of real data sets, financial transactions or credit card records; and IoT data or behavior data which repeatedly obtained low encryption time and decryption time as well as greater throughput than the prevalent Hybrid Approach (i.e. RSA-AES, DH-AES, Blowfish- AES etc.). The model, which lowers processing overheads and minimizes latencies while scaling with increased data sizes, solves various challenges associated with cloud data protection. These findings indicate it could be a solid And flexible solution for organizations who require both speed and secure. Next, we plan to experiment with the system in real industry settings as well as investigate extensions of the method using post-quantum algorithms and analyze how it withstands new security threats in order for it to be ready for future cloud platforms.

6. Statements and Declarations

Funding: This research received no external funding

Conflicts of Interest: The authors declare no conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

References

[1] Poppoppula, M., Prasad, M., R. R. Pbv, Pokkuluri, K. S., Babu, G. R., & Varma, C. P. (2023). A hybrid intelligent cryptography algorithm for distributed big data storage in cloud computing security. In Lecture Notes in Computer Science (pp. 637–648). https://doi.org/10.1007/978-3-031-36402-0_59

[2] Tuo, Z. (2023). A comparative analysis of AES and RSA algorithms and their integrated application. Theoretical and Natural Science, 25(1), 28–35. https://doi.org/10.54254/2753-8818/25/20240893

[3] Akter, A., Hossain, M. S., Muhammad, G., Hossain, M. S., & Alhamid, M. (2024). An optimized hybrid encryption framework for smart home healthcare: Ensuring data confidentiality and security. Internet of Things, 25, 101314. https://doi.org/10.1016/j.iot.2024.101314

[4] Bokhari, N., & Martínez Herráiz, J. J. (2024). A robust cloud security model leveraging a hybrid of cryptography and steganography. Authorea Preprints. https://doi.org/10.22541/au.171221467.72775845/v1

[5] Dutta, D. (2024). Secure storage on cloud using hybrid cryptography. International Journal of Advanced Research in Science, Communication and Technology. https://doi.org/10.48175/IJARSCT-18707

[6] Karanam, M., S. R. S., Chakilam, A., & Banothu, S. (2024). Performance evaluation of cryptographic security algorithms on cloud. E3S Web of Conferences, 391, 01015. https://doi.org/10.1051/e3sconf/202339101015

[7] Mondal, S., & Primajaya, A. (2024). The implementation of RSA algorithm in text messages encryption and decryption. Journal of Advanced Technology and Innovation, 8(6). https://doi.org/10.36040/jati.v8i6.12037

[8] Rahayu, A., Ardana, A. P., Pramudhita, C., Syafitri, D., & Siregar, R. Z. (2024). Comparison of algorithm RSA with algorithm Blowfish in data security application design. MATEC Web of Conferences, 3(2). https://doi.org/10.63893/matech.v3i2.224

[9] Shastri, V. H., & Pragathi, C. (2024). Data security using crypto bipartite graph theory with modified Diffie–Hellman algorithm. Journal of Supercomputing. https://doi.org/10.1007/s11277-024-11679-y

[10] Toma, N. I., Kashem, M. A., & Rana, S. (2024). A lightweight cryptographic algorithm for IoT devices based on hybrid architecture. In Proceedings of the 2024 International Conference on Advanced Engineering and Electronics Engineering. https://doi.org/10.1109/ICAEEE62219.2024.10561749

[11] Uriawan, W., Ramadita, R., Putra, R. D., & Siregar, R. I. (2024). Authenticate and verification source files using SHA-256 and HMAC algorithms. Preprints. https://doi.org/10.20944/preprints202407.0075.v1

[12] Buhari, B. A., Abdulkadir, H., Sulaiman, R., Ahmad, S. A., & Umar, M. M. (2025). Performance and security analysis of symmetric data encryption algorithms: AES, 3DES and Blowfish. International Journal of Advanced Networking and Applications. https://doi.org/10.35444/IJANA.2025.16404

[13] Chauhan, G. (2025). Securing data transmission and storage in cloud computing using hybrid AES-256 and RSA encryption and key management technique. International Journal of Science and Engineering Applications. https://doi.org/10.7753/IJSEA1403.1013

[14] Dutta, D. (2025). Secure storage on cloud using hybrid cryptography. International Journal of Advanced Research in Science, Communication and Technology. (Already listed above if 2024 → keep earlier)

[15] Elhoseny, M., Sangaiah, A. K., & Kim, H. W. (2025). A hybrid chaos-based cryptographic framework for post-quantum secure communications. arXiv Preprint. https://doi.org/10.48550/arXiv.2504.08618

[16] Kumar, A., & Sharma, S. (2025). Secure file storage on cloud using hybrid cryptography. International Journal of Scientific Research in Engineering and Management, 9(5), 1–9. https://doi.org/10.55041/IJSREM47521

[17] Qiang, L. (2025). Research on performance optimization and resource allocation strategy of network node encryption based on RSA algorithm. Journal of Computer Systems and Management. https://doi.org/10.13052/jcsm2245-1439.1415

[18] Shastri, V. H., & Pragathi, C. (2025). Data security using crypto bipartite graph theory (if listed again make consistent).

[19] Sikdar, S., Dutta, S., & Kule, M. (2025). On cryptanalysis of 3-DES using nature-inspired algorithms. International Journal of Computer Network and Information Security, 17(3), 54–71. https://doi.org/10.5815/ijcnis.2025.03.04

[20] Yang, R., & Gao, J. (2025). Enhancing financial data security in big data environments using AES-Blowfish and cloud-aided encryption techniques. International Journal of Information Security. https://doi.org/10.1142/S0218126625503396