INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Advancing Data Security in Cloud Computing: Introducing Secured Layered Technique for Data Security Approach (SLT-DSA), A Multi-Layered Security Framework

J Vinnarasi[1]*, D I George Amalarethinam[2]

1 Research Scholar, PG and Research Department of Computer Science, Jamal Mohamed College (Autonomous), Affiliated to Bharathidasan University, Tiruchirappalli, 620020, Tamil Nadu, India
2 Principal and Head, Associate Professor, PG and Research Department of Computer Science, Jamal Mohamed College (Autonomous), Affiliated to Bharathidasan University, Tiruchirappalli, 620020, Tamil Nadu, India

OPEN ACCESS

Received: 08-02-2025
Accepted: 15-03-2025
Published: 02-04-2025

Citation: Vinnarasi J, Amalarethinam DIG (2025) Advancing Data Security in Cloud Computing: Introducing Secured Layered Technique for Data Security Approach (SLT-DSA), A Multi-Layered Security Framework. Indian Journal of Science and Technology 18(11): 848-860. https://doi.org/10.17485/IJST/v18i11.273

* **Corresponding author**.

j.vinnarasicharles@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objective: The aim of this study is to introduce the Secured Layered Technique for Data Security Approach (SLT-DSA), a multi-layered framework designed to enrich data protection in cloud computing environments. This research aims to evaluate the effectiveness of SLT-DSA in addressing security challenges, improving data privacy, and ensuring scalability and performance in cloud-based systems. **Methods:** This paper has introduced the Secured Layered Technique for Data Security Approach (SLT-DSA), aimed at ensuring data security and integrity when outsourcing data in cloud environments. The enhancement of cloud data security has been demonstrated by employing a composite encryption algorithm and hash functions. The proposed SLT-DSA contains three layered techniques and secure hash algorithms, including scrambling with binary operation, dynamic key and secret key generation with encryption, and signature generation. Keys and encrypted data are kept in the AWS cloud. **Findings:** Three layered techniques are introduced in this work to provide secure data storage in the cloud environment. The scrambling and binary operation-based encryption technique is suggested to improve data confidence. The secure hash-based signature is used to check the integrity of data. This work presents the security analysis and experimental findings of SLT-DSA to demonstrate its superiority over existing methods. The suggested algorithm's experimental findings confirm its effectiveness with state-of-the-art encryption methods. **Novelty:** The results of the experiment has been improved through SLT-DSA framework. Compared to alternative encryption methods, the experimental outcomes of the suggested algorithm displayed a high degree of security, reducing cipher text size and decreasing encryption and decryption time.

Keywords: Hybrid Encryption; Secure Hash; Dynamic Key; Secret Key; Layered Technique; Matrix Scrambling

1 Introduction

Cloud computing is an emerging domain in computational sciences that uses the Internet to deliver quick and effective services. Many companies have moved their operations to the Cloud to take advantage of cloud computing. Because the Cloud offers data and computational services at request, it has proven to be an effective and adaptable solution for processing and storing data, adapting to the specific needs of many applications. There are still a lot of obstacles to be overcome in terms of privacy and data security. Even while the Cloud makes accessing and managing data storage easier, there is always a chance for illicit activity and access. Highly confidential information may be secretly stored on the cloud server. Thabit et al.[(1)] have made a detailed review on a new, efficient, and lightweight homomorphic cryptographic algorithm with two levels of encryption. The innovative, efficient, lightweight cryptographic technique is used in the first layer, and multiplicative homomorphic schemes are examined in the second layer to improve cloud computing security data. This method provides features for both symmetric and asymmetric cryptography. Viswanath et al.[(2)] discussed the encryption technique for massive data storage in several cloud storage systems. The user can save data across many cloud storage providers due to the multi-cloud storage environment. The author creates a secure architecture that limits insider threats. The framework includes processes for uploading, slicing, indexing, distributing, decrypting, retrieving, and merging data. It was created to safeguard large amounts of data before storing it across multiple clouds.

Sana et al.[(3)] presented a public key cryptosystem that uses a modified Rivest-Shamir-Adleman (RSA) algorithm for data encryption and decryption to store, retrieve, and transport data in cloud computing. The application of a modified RSA algorithm maintains data security in the cloud environment. A neural network is used for user authentication and recognition to improve user data security. Gupta et al.[(4)] recommended a hybrid chaotic-based DNA and multifactor authentication approach to improve cloud environment security. First, multimodal data are gathered from the data owner, and then the deflate compression technique is used to compress the data. To further improve data security, the data is subsequently encrypted utilizing hybrid chaotic-based DNA cryptography. In this hybrid algorithm, the encryption process uses a chaotic algorithm, while the key creation process uses DNA. A multifactor authentication mechanism is developed to prevent unauthorized users from accessing cloud data.

Maddila et al.[(5)] presented a unique data protection scheme for the Cloud Using the twofish encryption technique and the Bald Eagle Pelican Optimization (BEPO) algorithm for key generation. Initialization, registration, key generation, data encryption, authentication, validation, data sharing, and decryption are some of the processes in which the Twofish+BEPO_KeyGen is implemented. The data that needs to be sent to the Cloud is encrypted using the Twofish technique, and the BEPO algorithm generates the security key required for encryption.

Ramachandra et al.[(6)] suggested the Triple Data Encryption Standard (TDES) approach to secure massive amounts of data on the Cloud. By enlarging the keys in the Data Encryption Standard (DES) to withstand attacks and safeguard data privacy, the TDES methodology offers a comparatively easier solution. Koppaka et al.[(7)] proposed a hybrid cryptography algorithm which consists of the blowfish algorithm (BFA), the triple data encryption standard, and the advanced encryption standard (AES). First, user data is collected and uploaded to the cloud storage space. Next, a key is generated to facilitate the encryption and decryption of user data. The encryption and

decryption process uses the hybrid cryptography algorithm, which safely stores data and assists users according to their needs. Verma et al.[8] emphasized enhanced cloud security using hyperelectronic curves and biometrics (ECSHB). The ECSHB paradigm guarantees data security, privacy, and authentication maintenance in a cloud context. Using biometric and hyperelliptic curve cryptography (HECC) techniques, this method improves cloud resource preservation and data access security. ECSHB automatically lowers overall costs by offering a high level of security with less computing power.

Kumar et al.[9] discussed a secure cloud data access paradigm using quantum key distribution and attribute-based encryption (ABE). ABE is used in the study to encrypt user data and guarantee its security during transmission and storage. It is also possible to access cloud-based data using the quantum key. Rao et al.[10] have made a detailed review on Hybrid Elliptic Curve Cryptography (HECC) for public cloud security. The method uses a lightweight Edwards curve to generate keys. The produced private keys can be modified using the user's Identity-Based Encryption. The Advanced Encryption Standard encryption process is accelerated by using the key reduction method to shorten the keys further. Diffie Hellman key exchange is used to exchange the public keys.

Lalem et al.[11] discussed a novel digital signature mechanism for a family of encryption techniques. The strategy is divided into two sections: the secret key part and the public key part. To verify the validity of the signature, multiply these two components to reform the sender's public key once more and then compare the outcome with the message that has been decrypted. Data integrity is ensured by validating the decrypted message, and authenticity is verified using the signer public key.

Li et al.[12] proposed a AIVCI, a public data integrity verification technique, to verify data integrity in Cloud-IoT scenarios. Firstly, AIVCI can efficiently finish data auditing based on algebraic signature and homomorphic hash functions. To better safeguard IoT users' privacy and stop IoT data leaks, AIVCI also uses blind technology. Third, batch auditing is developed to meet practical requirements for Cloud-IoT scenarios and increase auditing efficiency. A new data structure called the Improved Divide and Conquer Table (ID&CT) was created to achieve effective data dynamics. The existing literature exposes the issue of privacy violations and security flaws in some of the cloud computing security models currently in use, which use encryption methods to store data. It has been shown that certain encryptions are insufficient to defend data and preserve privacy for data stored in the platform of cloud computing. This paper proposes a Secured Layered Technique for a Data Security Approach (SLT-DSA). The proposed SLT-DSA contains three layered techniques and secure hash algorithms. The first layer contains scrambling with a binary operation. The second layer includes dynamic key and secret key generation with encryption. The final layer has signature generation. The generated keys and encrypted shares are stored in cloud storage.

2 Methodology

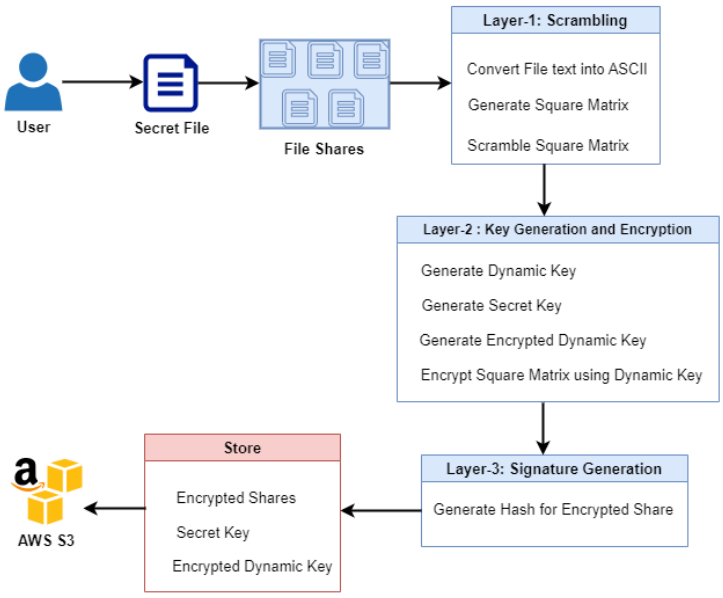


Fig 1. The proposed three-layered encryption architecture

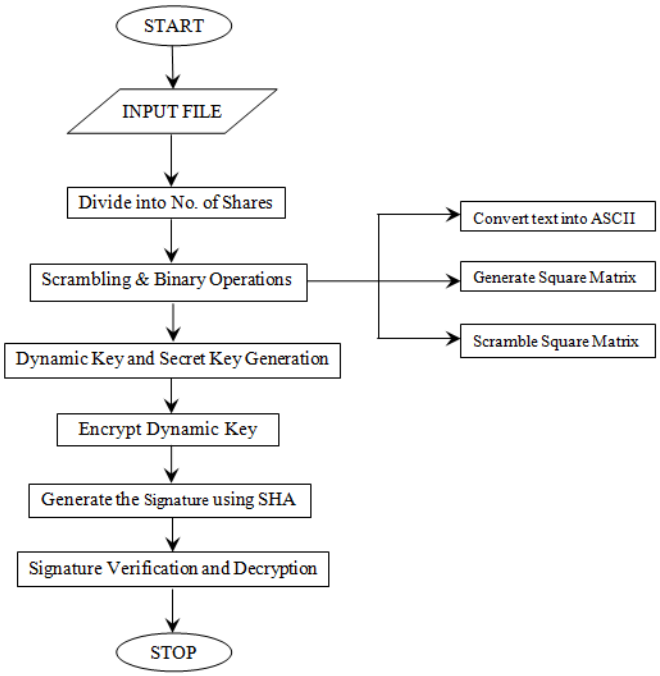


Fig 2. Proposed Work - Flow chart

Methodology section explains the proposed secured layered cloud data security technique. The Secured Layered Technique for Data Security Approach (SLT-DSA) is suggested to strengthen data security in cloud computing environments, achieving low processing and high speed. The proposed approach contains three layers: a scrambling layer, a key generation layer, and a secure hash layer. Initially, the input plain text is divided into the number of shares based on the size of the text. The scrambling layer contains three processes. The first process converts each character in the plain text into ASCII code, the next process is square matrix generation, and the final process is square matrix scrambling. The key generation layer includes the process of dynamic and secret key generation with dynamic key encryption and square matrix encryption. The secure hash layer generates the hash value for encrypted shares. Figure 1 shows the proposed three-layered encryption architecture. Figure 2 shows the flow chart of the secure layered technique.

2.1 Algorithm 1 Outlines the key steps involved in three-layered encryption

Algorithm-1 Three Layered Encryption

Input: Text File (TxtF)
Output: Encrypted File (EncF)
Step01: FCS = Extract File Content Size
Step02: Divide TxtF into the number of file shares based on FCS
Step03: For i = 1 to Number of Shares
Step04: Convert text into corresponding ASCII value
Step05: Generate Square Matrix
Step06: Scramble square matrix elements
Step07: End For
Step08: Generate Dynamic and Secret Key based on the randomly selected share
Step09: Generate Encrypted Dynamic Key
Step10: Encrypt each share using Dynamic Key
Step11: Generate a secure hash for each encrypted share
Step12: Store Keys and encrypted shares in the Cloud
Layer-1: Scrambling

The input text file is initially divided into a number of shares based on the file content size. Consider the sample text ' Secure layered encryption for cloud.' The size of the text is 36 bits, and the number of shares is 4 ($\lceil \frac{36}{10} \rceil = \lceil 3.6 \rceil = 4$).

The sample text is divided into four shares.

Share1 = 'Secure la'

Share2 = 'yered enc'

Share3 = 'ryption f'

Share4 = 'or cloud.'

Convert text into ASCII value.

Share1 = 83 101 99 117 114 101 32 108 97

Share2 = 121 101 114 101 100 32 101 110 99

Share3 = 101 101 99 101 114 100 32 110 121

Share4 = 111 114 32 99 108 111 117 100 46

Generate a square matrix and assign ASCII value from left to right in the matrix.

$$Share1 = \begin{bmatrix} 83 & 101 & 99 \\ 117 & 114 & 101 \\ 32 & 108 & 97 \end{bmatrix}$$

$$Share2 = \begin{bmatrix} 121 & 101 & 114 \\ 101 & 100 & 32 \\ 101 & 110 & 99 \end{bmatrix}$$

$$Share3 = \begin{bmatrix} 101 & 101 & 99 \\ 101 & 114 & 100 \\ 32 & 110 & 121 \end{bmatrix}$$

$$Share4 = \begin{bmatrix} 111 & 114 & 32 \\ 99 & 108 & 111 \\ 117 & 100 & 46 \end{bmatrix}$$

The square matrix position is interchanged using the scrambling technique.

$$Scrambled\ Share1 = \begin{bmatrix} 101 & 117 & 97 \\ 32 & 99 & 114 \\ 101 & 108 & 83 \end{bmatrix}$$

$$Scrambled\ Share2 = \begin{bmatrix} 101 & 101 & 99 \\ 101 & 114 & 100 \\ 32 & 110 & 121 \end{bmatrix}$$

$$Scrambled\ Share3 = \begin{bmatrix} 121 & 116 & 102 \\ 110 & 112 & 105 \\ 111 & 32 & 114 \end{bmatrix}$$

$$Scrambled\ Share4 = \begin{bmatrix} 114 & 99 & 46 \\ 117 & 32 & 108 \\ 111 & 100 & 111 \end{bmatrix}$$

Layer-2: Key Generation and Encryption

This section generates the dynamic and secret keys based on the scrambled share. Algorithm 2 explains the key generation algorithm.

2.2 Algorithm-2: Key Generation Technique

Input: Randomly selected scrambled share

Output: Dynamic Key (DK), Secret Key (SK), Encrypted Dynamic Key (EDK)

Step01: Get the randomly selected scrambled matrix

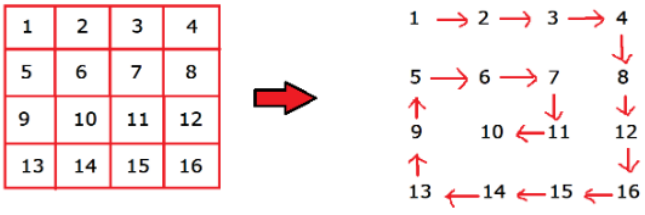
Step02: SP1 = Traverse scrambled matrix in spiral order.

Step03: SP2 = Traverse scrambled matrix in reverse spiral order.
Step04: Convert SP1 and SP2 into binary format.
Step05: Dynamic Key (DK)= Select 1st bit in each element.
Step06: Secret Key (SK) = Select the last bit in each element.
Step07: Encrypted Dynamic Key (EDK) = DK SK
Consider the randomly selected scrambled share

$$Scrambled\ Share3 = \begin{bmatrix} 121 & 116 & 102 \\ 110 & 112 & 105 \\ 111 & 32 & 114 \end{bmatrix}$$

Apply spiral traversal order. Spiral traversal in key generation enhances security by introducing non-linear complexity, making it resistant to predictable attacks. Unlike sequential methods, it distributes key values uniquely, reducing the risk of brute-force decryption. This approach ensures data scrambling and obfuscation, making unauthorized key reconstruction significantly harder.

SP1 = [121, 116, 102, 105, 114, 32, 111,110, 112]
SP2 = [112, 110, 111, 32, 114, 105, 102, 116, 121]
Example of spiral traversal of matrix



Convert the spiral traversal of matrix value into binary format
Bin_SP1 = 1111001, 1110100, 1100110, 1101001, 1110010, 0100000, 1101111
Bin_SP2 = 1110000, 1101110, 1101111, 0100000, 1110010, 1101001, 1100110
Select 1st bit in Bin_SP1.
Dynamic Key (DK) = 1 1 1 1 1 0 1
Select the last bit in Bin_SP2
Secret Key (SK) = 0 0 1 0 0 1 0
Encrypted Dynamic Key (EDK) = 1 1 0 1 1 1 1
Read the scrambling matrix row by row, convert each element into binary format, and find one's complement.

$$bin1 = \begin{bmatrix} 1100101 & 1110101 & 1100001 \\ 0100000 & 1100011 & 1110010 \\ 1100101 & 1101100 & 1010011 \end{bmatrix} \quad obin1 = \begin{bmatrix} 0011010 & 0001010 & 0011110 \\ 1011111 & 0011100 & 0001101 \\ 0011010 & 0010011 & 0101100 \end{bmatrix}$$

$$bin2 = \begin{bmatrix} 1100101 & 1100101 & 1100011 \\ 1100101 & 1110010 & 1100100 \\ 0100000 & 1101110 & 1111001 \end{bmatrix} \quad obin2 = \begin{bmatrix} 0011010 & 0011010 & 0011100 \\ 0011010 & 0001101 & 0011011 \\ 1011111 & 0010001 & 0000110 \end{bmatrix}$$

$$bin3 = \begin{bmatrix} 1111001 & 1110100 & 1100110 \\ 1101110 & 1110000 & 1101001 \\ 1101111 & 0100000 & 1110010 \end{bmatrix} \quad obin3 = \begin{bmatrix} 0000110 & 0001011 & 0011001 \\ 0010001 & 0001111 & 0010110 \\ 0010000 & 1011111 & 0001101 \end{bmatrix}$$

$$bin4 = \begin{bmatrix} 1110010 & 1100011 & 0101110 \\ 1110101 & 0100000 & 1101100 \\ 1101111 & 1100100 & 1101111 \end{bmatrix} \quad obin4 = \begin{bmatrix} 0001101 & 0011100 & 1010001 \\ 0001010 & 1011111 & 0010011 \\ 0010000 & 0011011 & 0010000 \end{bmatrix}$$

Applying logical XoR operation and converting it into character form

$$xbin1 = \begin{bmatrix} 1100111 & 1110111 & 1100011 \\ 0100010 & 1100001 & 1110000 \\ 1100111 & 1101110 & 1010001 \end{bmatrix} \quad char1 = \begin{bmatrix} g & w & c \\ " & a & P \\ g & n & Q \end{bmatrix}$$

$$xbin2 = \begin{bmatrix} 1100111 & 1100111 & 1100001 \\ 1100111 & 1110000 & 1100110 \\ 0100010 & 1101100 & 1111011 \end{bmatrix} \quad char2 = \begin{bmatrix} g & g & a \\ g & p & f \\ " & l & \{ \end{bmatrix}$$

$$xbin3 = \begin{bmatrix} 1111011 & 1110110 & 1100100 \\ 1101100 & 1110010 & 1101011 \\ 1101101 & 0100010 & 1110000 \end{bmatrix} \quad char3 = \begin{bmatrix} \{ & v & d \\ l & r & k \\ m & " & p \end{bmatrix}$$

$$xbin4 = \begin{bmatrix} 1110000 & 1100001 & 0101100 \\ 1110111 & 0100010 & 1101110 \\ 1101101 & 1100110 & 1101101 \end{bmatrix} \quad char4 = \begin{bmatrix} p & a & , \\ w & " & n \\ m & f & m \end{bmatrix}$$

Read matrix row by row,
Encrypted share1 = gwc”apgnQ
share2 = ggagpf”l{
share3 = {vdlrkm”p
share4 = pa,w”nmfm

Layer-3: Secure Hash Generation

Applications for hash functions include digital signatures, data integrity, password protection, message authentication, key derivation, pseudo-random number creation, and cryptography protocols. The message digest is a data set computed using hash function algorithms into a defined length cryptography hash.

The Secure Hash Algorithm (SHA) generates the hash (Message digest).The hash value of encrypted shares:

Share- 1
97C1E9A8A000236FD5A0C32B4746A21BD535C896
Share- 2
78A296A9C578F96C5C4493BE8D31BDF233D4F086
Share- 3
D0CD87994D28FA4809D1F121DD2DC17635A2C020
Share- 4
9C8EB76098783BB81B67F39D1FC4F4AA64349B6D

The generated encrypted dynamic key, secret key and encrypted shares are stored in the Cloud. The hash value of the encrypted shares is stored in the local machine (user side).

The encryption procedure is reversed during decryption. It originates from the user’s requests for data from the Cloud and their authentication process-verified login. When a User wants to decrypt the file stored in the Cloud, first download the encrypted dynamic key, secret key and encrypted shares. Decrypt the dynamic key using the secret key. Verify the hash of encrypted shares and locally stored shares. Decrypt the encrypted share using the dynamic key if both shares are equal. Data in cloud computing is stored remotely and may be kept in multiple places. Challenges with data integrity arise due to the complex traversal pattern, which may cause misalignment or errors in key reconstruction. The Secure Hash Generation algorithm has been used for checking the integrity of data.

3 Results and Discussion

Several security examinations and investigations are conducted to verify the recommended algorithm. The robustness of the suggested coding scheme is assessed based on the length of the cipher text, the encryption time, and the decryption time. To improve data security in cloud computing, hybrid cryptography method [13] tackles the main issues with cloud security by guaranteeing data confidentiality, integrity, and safe key management. Cloud service providers can successfully reduce the risks associated with illegal access and data breaches by putting hybrid cryptography into practice. The proposed work has been implemented as extension of two layered [14] encryption technique which includes scramble square matrix and binary logical operation. The suggested approach is compared to genetic and logical-mathematical (GLM) based [15] encryption algorithms based on the size of the plain text and the amount of time required for encryption and decryption. A novel data security

algorithm for cloud computing that integrates genetic techniques with logical-mathematical functions. The algorithm employs a two-layered encryption process. The proposed result has been compared with genetic & logical-mathematical (GLM) [15] and two layered [14] encryption technique for enhanced data security, improved cipher size, and reduced execution time compared to existing cloud security techniques. Determining the ratio of plain text to cipher text is crucial. Since this technology was created to store data in cloud files, cloud storage was considered when designing this approach to ensure that the data's size did not increase after encryption. Table 1 shows the comparison of cipher text size. Five different sizes of plain text (5, 10, 20, 30 and 40) were taken for the experiment. When the size of the plain text increased, the cipher text size is also increased for the GLM approach. The proposed SLT-DSA increased cipher text size by 1 for 5 and 10 kb files only. The proposed approach reduces cipher text size compared to the GLM approach.Figure 3 shows the comparison of plain text vs cipher size text.

Table 1. Size of Cipher Text Comparison

PlainText Size (KB)	GLM	SLT-DSA
5	15	6
10	30	11
20	60	20
30	90	30
40	120	40

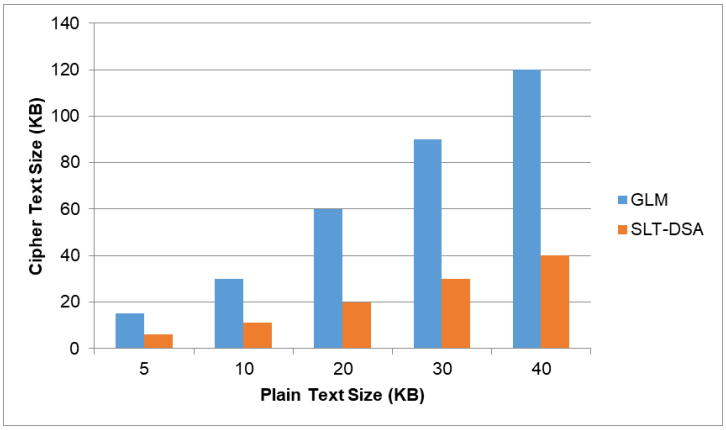


Fig 3. Comparison of Cipher Text Size

The algorithm that needs to transform plaintext into ciphertext is called the encryption time. The efficiency of an encryption technique is directly proportional to its encoding time; the faster the method, the higher its encryption efficiency. Table 2 shows the comparison of encryption time. Figure 4 compares encryption time for different sizes of plain text. The results indicate that the encryption process duration may increase linearly with a simple increase in the file size. Compared to the GLM technique, the suggested algorithm takes less time to encrypt the plain text. The average time to encrypt the plain text is 1.97 seconds for GLM and 0.4252 seconds for the proposed approach. The proposed SLT-DSA reduces 21.58% encryption time compared to GLM. Table 3 and Figure 5 compares encryption time for different character counts. When the character count is increased, the encryption time also increases.

Table 2. Encryption Time (Sec) Comparison

Plain Text Size (KB)	GLM	SLT-DSA
5	0.59	0.078
10	1.06	0.242
20	1.92	0.41
30	2.69	0.678
40	3.59	0.718
Average	1.97	0.4252

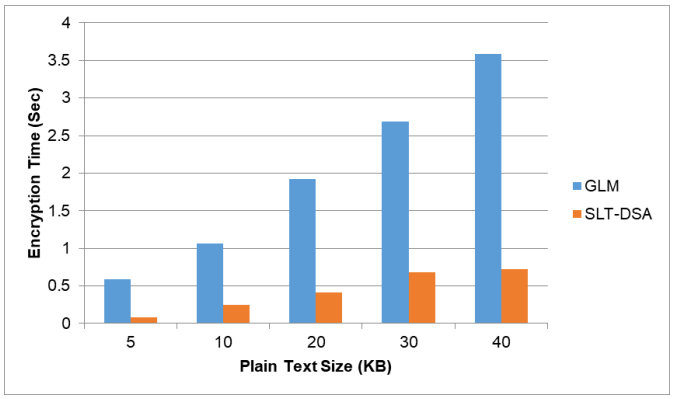


Fig 4. Comparison of Encryption Time

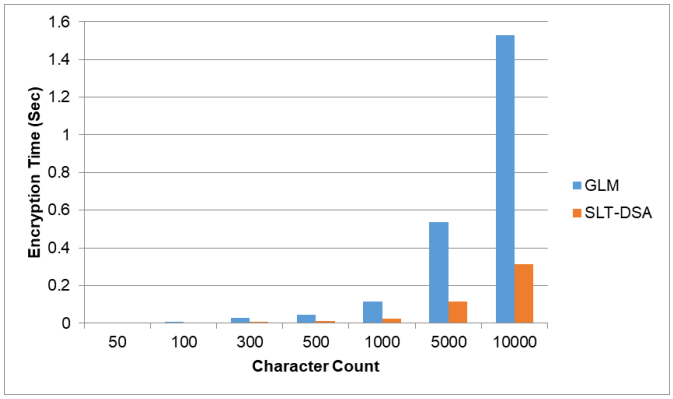


Fig 5. Comparison of Encryption Time for different character count

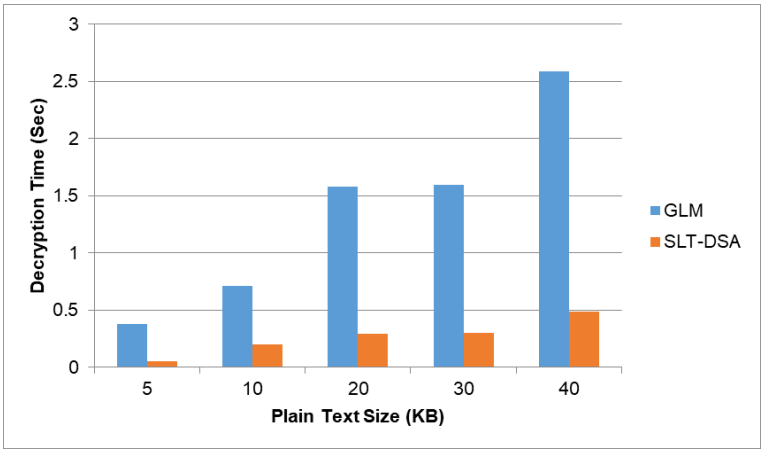


Fig 6. Comparison of Decryption Time

Table 3. Encryption Time Comparison for Different Character Counts

Character Count	GLM	SLT-DSA
50	0.0051	0.002
100	0.0092	0.004
300	0.030	0.007
500	0.046	0.010
1000	0.116	0.025
5000	0.537	0.113
10000	1.53	0.315

The duration required for the decrypting algorithm is often referred to as its temporal complexity. Table 4 shows the comparison of decryption time. Figure 6 compares decryption time for different sizes of plain text. The deciphering process takes less time than the decryption process. It shows that the suggested work has substantially less computational complexity. The average time to decrypt the cipher text is 1.3716 sec for GLM and 0.2662 sec for the proposed approach. The proposed SLT-DSA reduces 19.41% decryption time compared to GLM. When compared to GLM techniques, the suggested method decrypts faster. Therefore, the suggested model for safe communication can be implemented and is successful. Table 5 andFigure 7 show the decryption time comparison for different character counts. Table 6 shows the comparative analysis of two layered encryption technique with SLT-DSA method.

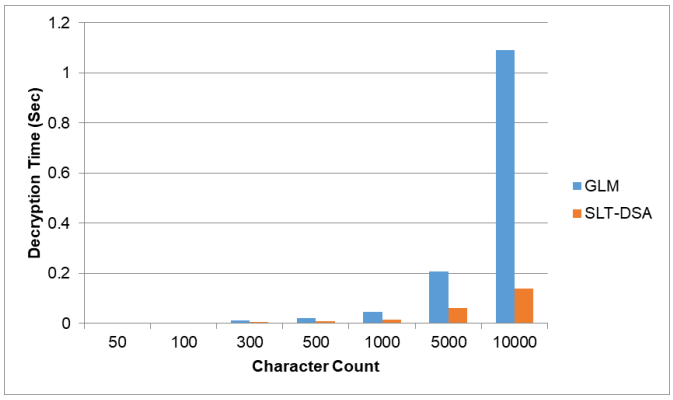


Fig 7. Comparison of Decryption Time for different character count

Decryption Time Comparison for Different Character CountsDecryption Time Comparison for Different Character Counts

Table 4. Decryption Time (Sec) Comparison

PlainText Size (KB)	GLM	SLT-DSA
5	0.381	0.054
10	0.712	0.199
20	1.581	0.293
30	1.594	0.302
40	2.59	0.483
Average	1.3716	0.2662

a. Implementation Details

The experimental steps of the suggested algorithm are explained in this section. The encrypted contents are kept in AWS, and Java (JDK-1.8) has been used to implement the algorithms. This research work uses Amazon S3 for storing the encrypted data. A public cloud storage resource offered by Amazon Web Services (AWS) Simple Storage Service (S3) platform is called an Amazon S3 bucket. Object-based storage keeps data in S3 buckets as discrete fragments called objects rather than files. File folders and Amazon S3 buckets are comparable in that they can be used for storing, retrieving, backing up, and accessing objects. Every

Table 5. Decryption Time Comparison for Different Character Counts

Character Count	GLM	SLT-DSA
50	0.00204	0.001
100	0.00328	0.002
300	0.0118	0.004
500	0.0208	0.009
1000	0.0446	0.015
5000	0.208	0.062
10000	1.09	0.14

Table 6. Shows the comparisonbetween Two Layered Encryption technique and SLT-DSA Algorithm

Size of Plain Text (bytes)	Layered Encryption			SLT-DSA		
	Cipher Text Comparison	Encryption Time Comparison	Decryption Time Comparison	Cipher Text Comparison	Encryption Time Comparison	Decryption Time Comparison
112	128	994	892	110	855	801
2305	2352	2057	1337	2052	1998	1227
7894	8192	10155	1586	8002	10098	1475
153422	153664	96673	2179	150112	95522	2078

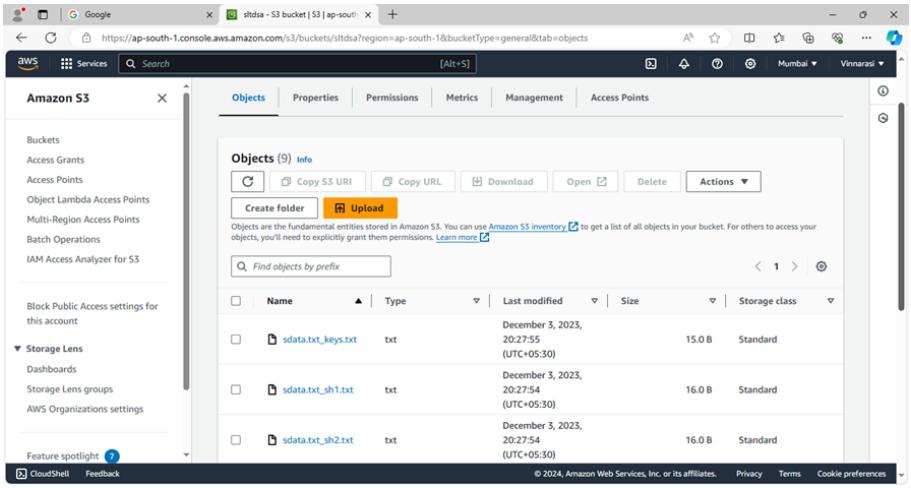


Fig 8. The objects stored in the sltdsa bucket

object consists of three primary parts: its content, or data; its unique identifier; and its descriptive metadata, which includes the object's name, URL, and size. Figure 8 shows the objects stored in the sltdsa bucket. Figure 9 shows the overview of the stored object (sdata.txt_keys.txt). The object's properties include owner, AWS region, date, size of object and URL. And also Figure 10 shows the downloaded objects from the sltdsa bucket. The user can save the downloaded objects on their own devices from anywhere.

Object storage services from Amazon Simple Storage Service provide industry-leading scalability, data accessibility, protection, and efficiency. Concerning cloud-native apps, mobile apps, data lakes, and other use cases, users of all sizes and sectors can store and safeguard any volume of data. Organizations may optimize expenses, organize data, and establish fine-tuned access restrictions to suit specific business, administrative, and compliance requirements with affordable storage classes and user-friendly management tools.

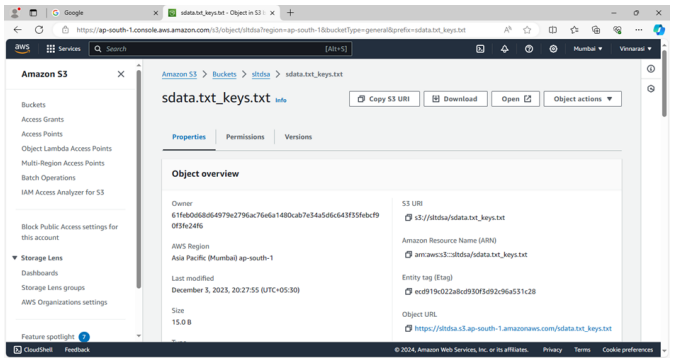


Fig 9. The overview of the stored object (sdata.txt_keys.txt). The object's properties include owner, AWS region, date, size of object and URL

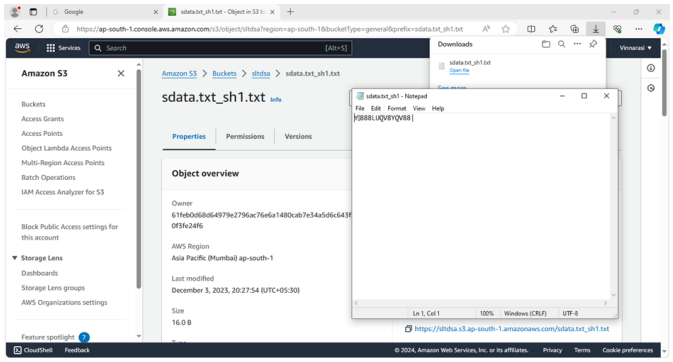


Fig 10. The downloaded objects from the sltdsa bucket. The user can save the downloaded objects on their own devices from anywhere

4 Conclusion

Data security is always the main obstacle to the adoption of cloud computing. Although various methods and algorithms guarantee data security, a gap must be filled. This study proposes a secure layered encryption technique. The suggested approach uses a cryptographic method based on scrambling with binary operation and secure hash creation to increase the data's level of secrecy during encryption. Three layers of encryption are used in the technique, and new dynamic key generation increases security. The suggested technique produces better results regarding encryption, decryption time, and ciphertext size. The proposed techniques have been checked for the achievement of CIA(Confidentiality, Integrity and Availability) Attainment. The suggested solution, which encrypts all shared contents and keys using the secured layered technique, has produced confidentiality. One of the most important risks to data security is data integrity. Malicious users can alter the data by evading the authentication procedure. In this scheme, the secure hash algorithm achieved data integrity. Availability refers to making information accessible and available to users whenever and wherever needed. Since the offered cryptographic algorithms are hypothetical, they can be applied anytime. To develop a more robust privacy protection system to guarantee that private information in shared data, and Use DNA-based cryptography for secure data encryption. The complexity of the suggested algorithm can be decreased to meet the challenge of the requirements better. More reliable encryption systems can be created with this computational approach.

5 Contributory statement

Formulation of the research problem, Algorithm & cryptographic method: J. Vinnarasi, Research Guidance: Dr. D. I. George Amalarethinam.

References

1) Thabit F, Can O, Alhomdy S, Al-Gaphari GH, Jagtap S. A Novel Effective Lightweight Homomorphic Cryptographic Algorithm for data security in cloud computing. *International Journal of Intelligent Networks*. 2022;3:16–30. Available from: https://www.sciencedirect.com/science/article/pii/S2666603022000033.
2) Viswanath G, Krishna PV. Hybrid encryption framework for securing big data storage in multi-cloud environment. *Evolutionary Intelligence*. 2021;14:691–698. Available from: https://link.springer.com/article/10.1007/s12065-020-00404-w.
3) Sana MU, Li Z, Kiren T, Liaqat HB, Naseem S, Saeed A. A Secure Method for Data Storage and Transmission in Sustainable Cloud Computing. *Computers, Materials & Continua*. 2023;75(2):2741–2757. Available from: https://www.researchgate.net/publication/369733856_A_Secure_Method_for_Data_Storage_and_Transmission_in_Sustainable_Cloud_Computing.
4) Gupta M, Ahuja L, Seth A. Security enhancement in a cloud environment using a hybrid chaotic algorithm with multifactor verification for user authentication. *International Journal of Computers and Applications*. 2023;45(11):680–696. Available from: https://www.tandfonline.com/doi/abs/10.1080/1206212X.2023.2267839.
5) Maddila SK, Vadlamani N. A Novel Efficient Hybrid Encryption Algorithm Based on Twofish and Key Generation Using Optimization for Ensuring Data Security in Cloud. *Journal of Information & Knowledge Management*. 2024;23. Available from: https://doi.org/10.1142/S0219649223500624.
6) Ramachandra MN, Rao MS, Lai WC, Parameshachari BD, Babu JA, Hemalatha KL. An efficient and secure big data storage in cloud environment by using triple data encryption standard. *Big Data and Cognitive Computing*. 2022;6(4). Available from: https://doi.org/10.3390/bdcc6040101.
7) Koppaka AK, Lakshmi VN. An Efficient and Secured Big Data Storage in a Cloud-based Environment Using Hybrid Cryptography Algorithm and Rivest, Shamir, Adleman Algorithm. *International Journal of Intelligent Engineering & Systems*. 2024;17(1). Available from: https://oaji.net/articles/2023/3603-1703735439.pdf.
8) Verma G, Kanrar S. A novel model to enhance the data security in cloud environment. *Multiagent and Grid Systems*. 2022;18:45–63. Available from: https://content.iospress.com/articles/multiagent-and-grid-systems/mgs220361.
9) Kumar A, Verma G. Secure Cloud Data Access: Unifying Quantum Key Distribution and Attribute-Based Encryption for Enhanced Data Protection. *SN Computer Science*. 2023;4(6). Available from: https://link.springer.com/article/10.1007/s42979-023-02285-z.
10) Rao BR, Sujatha B. A hybrid elliptic curve cryptography (HECC) technique for fast encryption of data for public cloud security. *Measurement: Sensors*. 2023;29. Available from: https://www.sciencedirect.com/science/article/pii/S2665917423002064.
11) Lalem F, Laouid A, Kara M, Al-Khalidi M, Eleyan A. Novel Digital Signature Scheme for Advanced Asymmetric Encryption Techniques. *Applied Sciences*. 2023;13(8). Available from: https://www.mdpi.com/2076-3417/13/8/5172.
12) Li Y, Li Z, Yang B, Ding Y. Algebraic Signature-Based Public Data Integrity Batch Verification for Cloud-IoT. *IEEE Transactions on Cloud Computing*. 2023;11(3):3184–3196. Available from: https://ieeexplore.ieee.org/document/10100872.
13) Reddy KK, Chadha AR, Nikhil PS, Sountharrajan S. Hybrid Cryptography Techniques for Data Security in Cloud Computing. In: 2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT). 2024;p. 1836–1842. Available from: https://ieeexplore.ieee.org/document/10486794.
14) Amalarethinam DG, Vinnarasi J. Enhancing the data security of cloud computing critical systems using layered encryption technique. *International Journal of Critical Computer-Based Systems*. 2024;11(3):194–215. Available from: https://www.inderscienceonline.com/doi/10.1504/IJCCBS.2024.143209.
15) Thabit F, Alhomdy S, Jagtap S. A new data security algorithm for the cloud computing based on genetics techniques and logical-mathematical functions. *International Journal of Intelligent Networks*. 2021;2:18–33. Available from: https://www.sciencedirect.com/science/article/pii/S2666603021000063.