SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V12I1P110

Volume 12 Issue 1, 129-139, January 2025

Original Article

A Reliable Environment with Extensive Advanced Encryption Standard Algorithm in Cloud Computing

E. Geetha Rani[1], Chetana Tukkoji[2]

[1,2]*Department of CSE, GITAM University, Bengaluru, India.*

[1]*Corresponding Author : grani@gitam.in*

Received: 14 November 2024 Revised: 20 December 2024 Accepted: 10 January 2025 Published: 30 January 2025

Abstract *- Web-based Cloud Computing is widely used to store data. Most firms who are shifting to the cloud find it cost-effective. Despite its popularity, it has presented several security concerns to its users. Cloud Computing data security has become a critical issue that requires prompt response. This paper proposes an appropriate usage of AES and an effective cloud-based data storage solution. The proposed data security solution is implemented by improving the standard AES algorithm. Furthermore, the proposed techniques use de-duplication to save storage space for user data. The Extensive Advanced Encryption Standard (EAES) methodology was applied, and the results were superior to the conventional Advanced Encryption Standard method. The research tries to tackle the issues stated above in stages. Initially, a broad library of strategies addressing the data security problem is explored, and their limitations are noted. The Extensive Advanced Encryption normal technology is used, yielding considerable results compared to the normal Advanced Encryption Standard approach.*

Keywords *- EAES, Security, Cloud Computing, AES.*

1. Introduction

Cloud Computing setups involve multiple remote servers operating together over the internet and networks. Users save or retrieve information in files that are stored centrally. The networks provide them online access via cloud computing services such as IaaS, PaaS, and SaaS. This changing paradigm has created a slew of new challenges [1], the most pressing of which is data security. Data transferred via the internet is vulnerable to attacks, making it a weak point for cloud operations. It is a critical issue that must be handled based on users' privacy and security concerns. Data segregation and session hijacking have grown commonplace in network environments. Furthermore, Cloud environments comprise genuine physical resources handled in a virtual environment, which presents significant security challenges. The challenges and issues encountered in existing Cloud contexts are discussed below.

IT areas have been vulnerable to attacks, particularly in open environments, placing data in danger. This problem becomes magnified when the data is kept in a separate system or geographical area and accessed over the internet. Thus, data privacy and security have become key concerns for users in Cloud contexts. This aspect has been studied both commercially and academically for a long time. CC's future development lies in mitigating this issue as commercial and non-commercial organizations move towards the cloud [2]. Securing data in the cloud involves both hardware and software. User and industry concerns on data security can be discussed as several factors, starting with data integrity. The confidentiality of information is a critical component in protecting users' private or confidential data. Clouds utilize authentication and access constraints to ensure this factor. TCCs that address confidentiality are deemed trustworthy and reputable. Despite widespread trust in CSPs, it is hard to eliminate potential security concerns from insiders. Simple encryption and key management do not address the issue since attacks grow more difficult when they interfere with network searches or perform simultaneous alterations. Only fine-grained authorizations can aid in recovery from such difficulties. One significant advantage of clouds is data availability. In the event of hard disk loss or natural disasters such as fire or network outages, users' data may not be completely retrieved, leaving them with just the CSP's guarantee. Furthermore, the fact that user data exists in multiple geolocations raises severe concerns about data security because CSPs are subject to local legal restrictions. CSPs must emphasize to clients these difficulties to gain their trust by ensuring data security and explaining local regulations to potential customers. Sharing data storage locations/relocations, availability, and fees might help develop CC users' confidence. Even while clouds store user data openly, users have fewer control options. This refers to personal information that is hidden and only available to those whom people trust. Privacy can be described in terms of three elements: when, how, and extent. When indicated, prior

knowledge relevant to the future may be revealed. This implies that a person may share information with friends and refrain from receiving automatic warnings. Extent refers to information that has been reported as ambiguous. Cloud privacy implies that sensitive data must be protected against adversaries and leaks.

1.1. Dedupe

Data de-duplication reduces duplicates in files, hence reducing space in CC storage. CSPs execute de-duplication in the background while other processes are running. This is commonly done during data replication, vault backups, or file transfers between clouds [3-5]. De-duplication can also be considered an intelligent file compression technique because it reduces data storage space. The procedure ensures that only one instance of a file exists in the cloud. When users save the same file again, only a pointer to the new file is kept, and no new copy is created. Each file or chunk is assigned a unique hash number (at the block level) using hash algorithms such as MD5 or SHA1. The produced hash number is compared with the index's current hash numbers. If this happens, the new hash number and data are preserved, and the information is not saved elsewhere. Chunks of data identified with the hash may repeat, or a hash collision may occur, resulting in data loss because only one chunk is retrieved using the hash. Figure 1 explains the deduplication operation for three users. Users submit their files to the server for storage, and the files X1, X2, and X3 are replicated and de-duplicated. One important advantage of de-duplication lies in speeding up backups or recovery of files in the cloud. Data de-duplication processes can be classed based on data unit, disk and location placement, as stated below. Data de-duplication is performed at both the file and block levels.

In the former, two files' hash values are compared; if they match, one is de-duplicated. In block-level de-duplication, chunks are produced by breaking up the file contents. The chunks created can be of fixed or variable length [6-9]. As the name says, fixed-size chunking breaks the file into equal-sized chunks. It is faster than other chunking techniques. However, it suffers from the "Boundary Shift" problem whenever the data changes. Variable length blocks improve data de-duplication throughput. Disk placement De-duplication is dependent on how data is saved on disk. Two strategies are used: forward and backward reference. Forward reference preserves fresh data chunks while creating pointers to older data chunks. Backward reference data pieces are heavily fragmented.

Location-based de-duplication is divided into two forms based on the source and target. On the client side, source-based deduplication is carried out. The data is de-duplicated before being sent to the storage server. This conserves both network bandwidth and storage space. Target-based de-duplication occurs on the server side, and the client is unaware of the process. There is no overhead for the client. Targeted de-duplication saves storage space but not network bandwidth. Although it costs less, the encryption process takes longer. ECC elliptical curves have finite values and exhibit symmetry along the graph's x-axis. ECC's 512-bit key is equal to RSA's 15,360-bit key. Prime number factorization is the foundation of RSA, a product of two huge primes. It is a complicated method to decrypt the assigned prime numbers, which makes it secure. RSA decryption can be expensive and time-consuming, even on huge devices. RSA can be highly useful at higher levels of security, but it is getting increasingly wasteful. DES/3DES information three times.

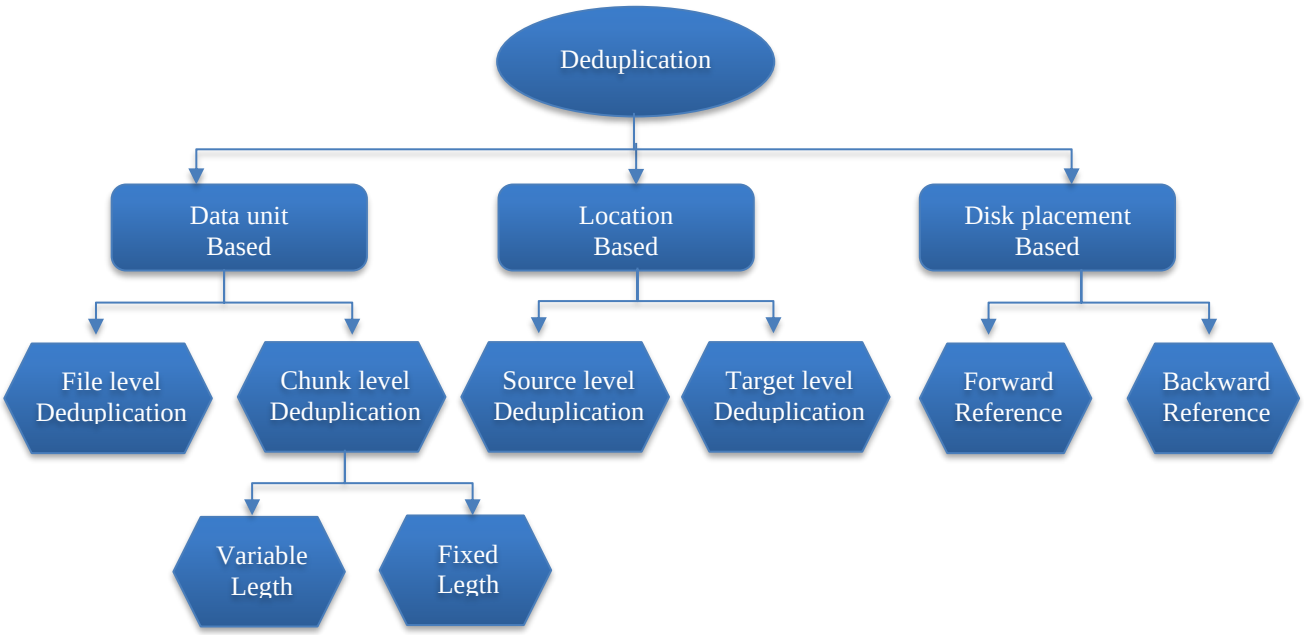


Fig. 1 Categorization of Dedupe

The symmetrical encryption method is triple DES, and the key is kept private. In CBC (Cipher Block Chaining) mode, many data encryptions fail. AES: This symmetric block cipher, also known as the Rijndael cipher, was introduced in 2002, and the ANIST issued a study in 2001 following brute-force attacks that compromised DES keys. The encryption and decryption block sizes are 128 bits. It is mostly used to protect sensitive government information.

2. Literature Survey

The biggest issue with employing secret keys in CC is that their numbers increase proportionally as the number of files increases. It becomes inconvenient when users must note or remember many concealed key values. Convergent encryptions are inefficient and pose complications, but AES encryptions, while considered safe, require a long time to perform because each encryption block only contains 128 bits. Using the same privilege key for data storage and retrieval may enable hackers or attackers to guess the access key easily. These safety concerns have been a barrier to limiting industrial Cloud applications [11].

The cloud protection solution is then proposed using the two-way password concept. After the user is authorized, this method produces two passwords. The biggest disadvantage of this strategy is that, for no apparent reason, the user must trust the cloud provider, and the mechanism is difficult. A privilege key is used in modern cloud protection techniques to store and retrieve data. Data is sent with the privilege key for storage, encrypted with convergent encryption, and transferred to a hybrid cloud. De-duplication is a valuable and promising strategy for managing cloud data, as mentioned in earlier sections. These safety concerns have been a hurdle to limiting industrial Cloud applications [12].

Unlike standard de-duplication approaches, which extract data from existing information, this methodology removes duplicate data from new data [13]. This highly dependable approach can be used to reduce the number of bytes delivered during network transfers. Cloud storage provides a low-cost way for personal computing devices to secure data. Cloud enables centralized cloud administration, which improves dependability and cost-effectiveness.

This provides quick disaster recovery backup storage, which is still critical to data backup. Backup datasets for cloud-based IT resources are highly redundant. There is certainly plenty of potential for improvement in terms of cloud storage performance. MLE (Message-Locked Encryption), introduced in [14], is a novel cryptographic approach that extracts both encryptions and decryptions from source material. Another current de-duplication approach, RevDedup, employs the reverse de-duplication idea. Unlike standard de-duplication approaches, which extract data from existing data, this methodology removes duplicate data from new data [15].

3. Proposed System

This proposed system saves and retrieves data in the cloud in two independent steps. The file is de-duplicated to eliminate duplications in the storage. The tag key of the upload file is matched while saving the file, and if duplicates are detected in storage, a message warns the user. If the tag keys of the uploaded and saved tags do not match, the file is encrypted with a 20-byte unique SHA hash. Then, an EAES (Extensive AES) encrypts this hash key and uploads the file before saving it. While retrieving an uploaded file, the user's `login` credentials are authenticated first, and if the authentication does not fail, the user gets a list of files as a view. On choosing a file to be downloaded or viewed, the user receives a decrypted hash key and a decrypted file. Encryptions in this proposed methodology are implemented based on the Rijndael algorithm, invented by Belgian scientists Vincent Rijimen and Joan Daemen [16]. The proposed EAES is also defined as a symmetric block cipher but uses 256-bit blocks for encryption with 128,192- or 256-bit keys. Simplicity of EAES is its key factor while its speed of operations saves memory. One significant factor of EAES is using the same hash key for encryptions and decryptions. The proposed architecture is organized on rational blocks that act in tandem. Figure 3 depicts the architecture's components, which are further discussed below. The CC environment is a public cloud that is cost-effective for its customers. To save expenses, the suggested approach stores data on public clouds. The user's files are encrypted and kept in this public cloud.

The initial stage is to produce tag keys for user-uploaded files, which are then de-duplicated. Uploads are encrypted and hence safe, and the de-duplication procedure ensures no duplicates. There is no fixed value for tag keys because they are generated based on file content and thus unique to each file. File tag keys are employed in cloud-based search operations to check for duplicates or redundancies. Every uploaded file is assigned to a file tag key, which is then compared to a database of tag keys to detect duplication. Only files without a match in the tag key dataset are saved. As a result, generations of similar hash values are avoided, and the amount of storage space in the cloud is reduced because only one copy of any file exits the cloud. SHA is the primary algorithm for hash key generation. This hash is used for both file storage and retrieval. Encryptions in this proposed framework use EAES, which is based on AES. Simple text is converted into ciphertext. When a file is saved, both the uploaded file and the SHA hash key are encrypted. This assures data safety in the public cloud, which is subject to threats. Decryptions are likewise safe because the keys and files are decrypted with EAES. Only authenticated users can post or download files, discouraging unlawful entry into the zone. The proposed framework's decryption is also performed using EAES, ensuring that the secretive key may only be accessed by the system with consent from the CSP and the user.

4. Methodology

As cyber-attacks become more sophisticated and frequent, it's more important than ever to ensure that our electronic data is securely encrypted. In the past, older encryption methods were sufficient for protecting sensitive information, but they no longer meet today's security needs. That's where AES encryption comes in.

4.1. Choosing The New AES Algorithm

- IBM Research created MARS, a block cipher.
- RSA Security designed the RC6 block cipher.
- Rijndael: Created by Belgian cryptographers Joan Daemen and Vincent Rijmen.
- Serpent was created by Ross Anderson, Eli Biham, and Lars Knudsen.
- Twofish, developed by Counterpane Internet Security. After the algorithms were tested and reviewed, the Rijndael algorithm was chosen as the proposed AES in 2000, and it later became a federal government standard.

4.2. Attacks on AES Encryption

Here's how to get the best possible protection out of AES encryption and minimize risk:

- Use long, hard-to-guess passwords
- Use Multifactor Authentication (MFA)
- Utilize a password manager
- Training employees in security best practices
- Use firewalls and anti-malware tools

4.3. The Advantages of AES

Some of the perks of using AES encryption include:

- Simple implementation and Fast encryption and decryption
- Robust security and Versatile key lengths
- Less memory-demanding than other types of encryptions

4.4. Algorithm for the Proposed Framework

Step : 1 Start with the input file.
Step : 2 Tag and generate a hash key to ensure deduplication.
Step : 3 Apply EAES encryption to the input file with the created key.
Step : 4 Upload the encrypted file to the server using EAES encryption.
Step : 5 After uploading the file, it is downloaded to the server and decrypted with the hash-derived key to recover the original file.
Step : 6 The system's performance is influenced by the synergistic effects of de-duplication and EAES encryption, which optimize storage space while improving cloud server security.

In the architecture, thin line arrows represent the system's path in data storage and retrieval for a user, whereas thick line arrows describe the system's internal paths for data storage, encryption, decryption, and file retrieval.

The module authenticates a cloud user requesting service with a valid login/password and only allows admission into the cloud after the user has been verified.

In the proposed design, this module serves as a user's primary entry point to the cloud. The suggested system produces hash keys for files depending on their contents, ensuring that each file's hash key is unique. Furthermore, these keys are produced automatically during file uploads and

The system uses keys to de-duplicate the file. This technique of reducing storage by eliminating file duplicates is based on the hash key created during a user's file uploads. The created file hash keys are examined in its key database for matching values; if any are found, the file is de-duplicated; otherwise, no earlier copies of the file are maintained.

The proposed EAES uses a network of replacement permutations for encryption. EAES considers 256 bits of plain text from an input array of characters as a block when converting them into cipher. This lowered encryption execution times by 50% compared to standard AES, which separates text into 128-bit blocks for encryption.

Table. 1. Comparison of RSA and AES for different Key Sizes

Point of Comparison	RSA	AES	AES-128	AES-256
Type	Asymmetric	Symmetric	128 bits	256 bits
Key Size	1024, 2048, 4096 bits	128, 192, or 256 bits	10	14
Speed	Slower	Faster	Secure	Very secure
Security	Less secure	More secure	More secure	More secure

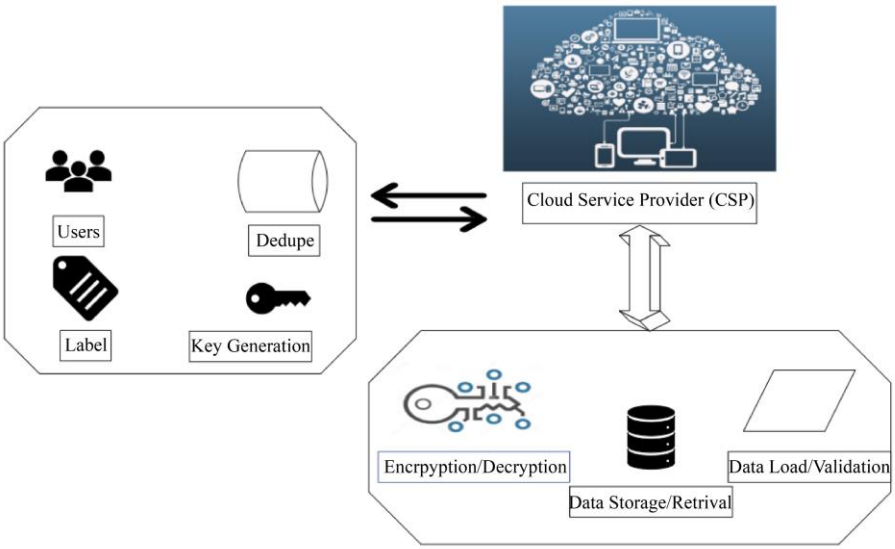


Fig. 2 illustrates the overall device architecture and explains each module

4.5. Encryption Process

In an encryption mode, a preliminary key is applied to the input values at the start. The technique is then repeated multiple times. Each round follows the procedure outlined below. A non-linear byte replacement in which each state byte operates independently via an S-box known as the Sub-Byte convention level.

The S-box is a pre-calculated substitution table that has 256 numbers (0-255) and their matching output values. During the Sub-Bytes stage, each byte in the state matrix a(i,j) is replaced with a Sub Byte bi,j via an 8-bit replacement box known as the Rijndael S-box. The transformation of Sub Bytes is based on the Galois Field Inverse operation GF (28) for achieving non-linearity. Additionally, the use of Galois Field, based on basic algebra, helps in avoiding attacks. The S-box is created by combining the inverse function with the transformation of an invertible description.

Shift Rows: During the Shift Rows transformation phase, the state's rows are regularly shifted at different offsets. Row 0 remains in position, while rows 1, 2, and 3 move one, two, and three bytes to the left.

Mix Column: As illustrated in Figure 4-6, there is a transposition of linear transformation from the Mix Column operations used to input the 4-byte in each column. The goal of this stage is to convert 4-bytes into 4-byte inputs and outputs, with each input byte affecting all 4-byte outputs. Each column is changed using fixed matrix operations, which include the multiplication and addition of entries, as shown in Figure 2. The addition is simply XOR. Multiplication is an irreducible polynomial modulus. In the Mix Column approach, each column is seen as a polynomial over GF before being multiplied by a fixed matrix polynomial c(x), which multiplies each column.

Add Round key: In this phase, Rijndael's key schedule generates a sub-key from the main key in each round, which is then integrated into the key state. The size of the sub-keys is equal to the size of the state. The integration procedure employs bitwise XORR to integrate sub-key and state bytes.

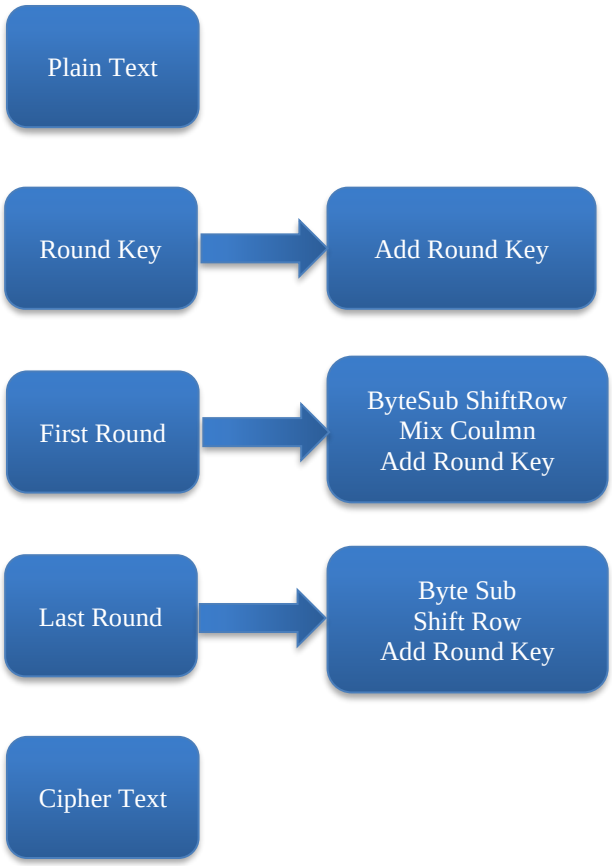
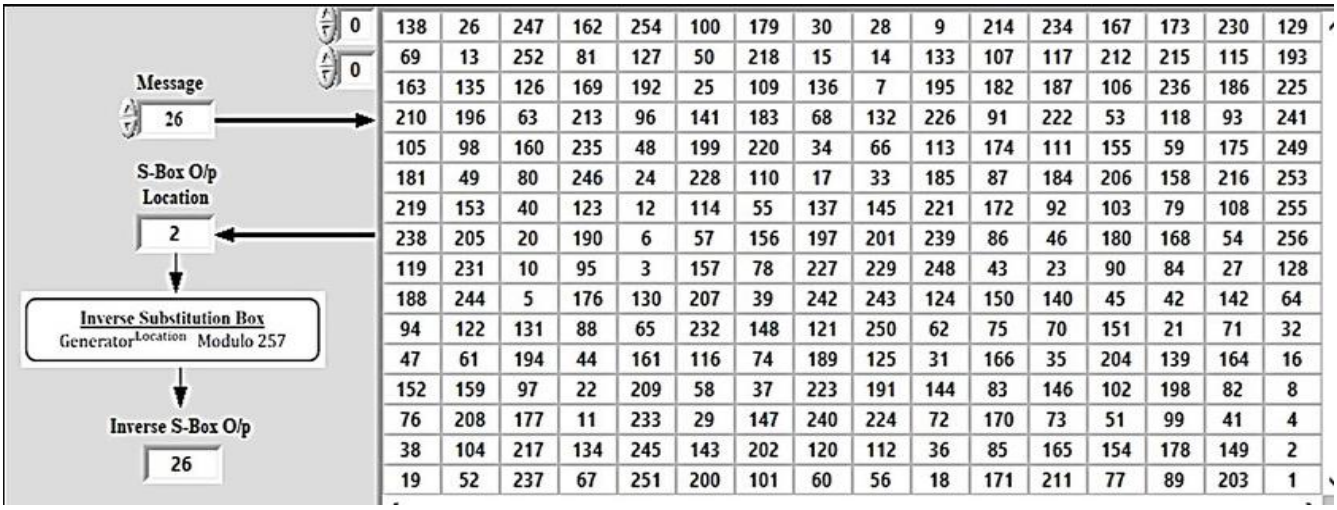


Fig. 3 Extensive advanced encryption is performed using a standard algorithm



138	26	247	162	254	100	179	30	28	9	214	234	167	173	230	129
69	13	252	81	127	50	218	15	14	133	107	117	212	215	115	193
163	135	126	169	192	25	109	136	7	195	182	187	106	236	186	225
210	196	63	213	96	141	183	68	132	226	91	222	53	118	93	241
105	98	160	235	48	199	220	34	66	113	174	111	155	59	175	249
181	49	80	246	24	228	110	17	33	185	87	184	206	158	216	253
219	153	40	123	12	114	55	137	145	221	172	92	103	79	108	255
238	205	20	190	6	57	156	197	201	239	86	46	180	168	54	256
119	231	10	95	3	157	78	227	229	248	43	23	90	84	27	128
188	244	5	176	130	207	39	242	243	124	150	140	45	42	142	64
94	122	131	88	65	232	148	121	250	62	75	70	151	21	71	32
47	61	194	44	161	116	74	189	125	31	166	35	204	139	164	16
152	159	97	22	209	58	37	223	191	144	83	146	102	198	82	8
76	208	177	11	233	29	147	240	224	72	170	73	51	99	41	4
38	104	217	134	245	143	202	120	112	36	85	165	154	178	149	2
19	52	237	67	251	200	101	60	56	18	171	211	77	89	203	1

Fig. 4 Substitution box

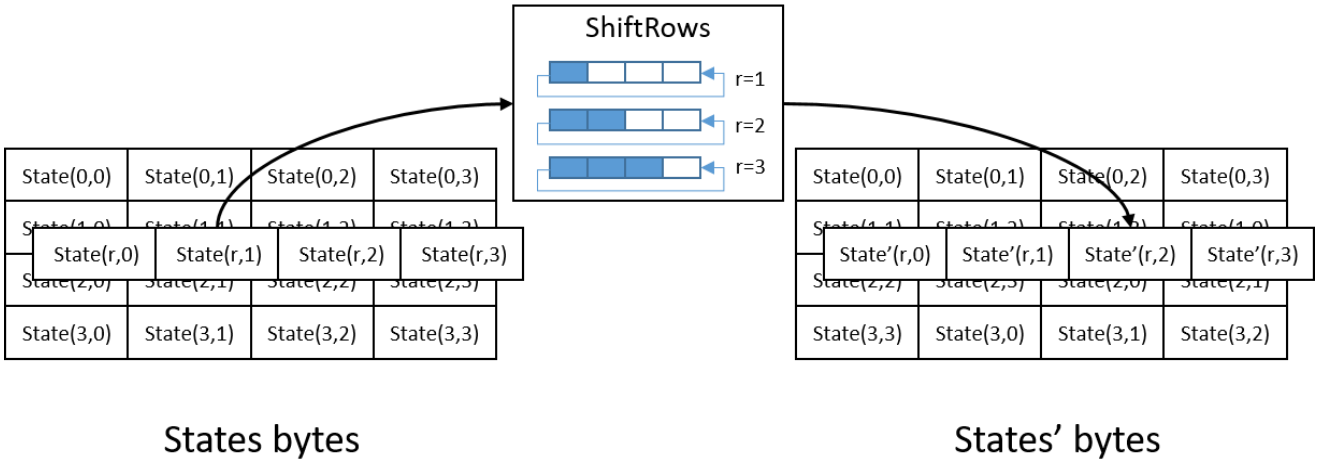


Fig. 5 Shift rows

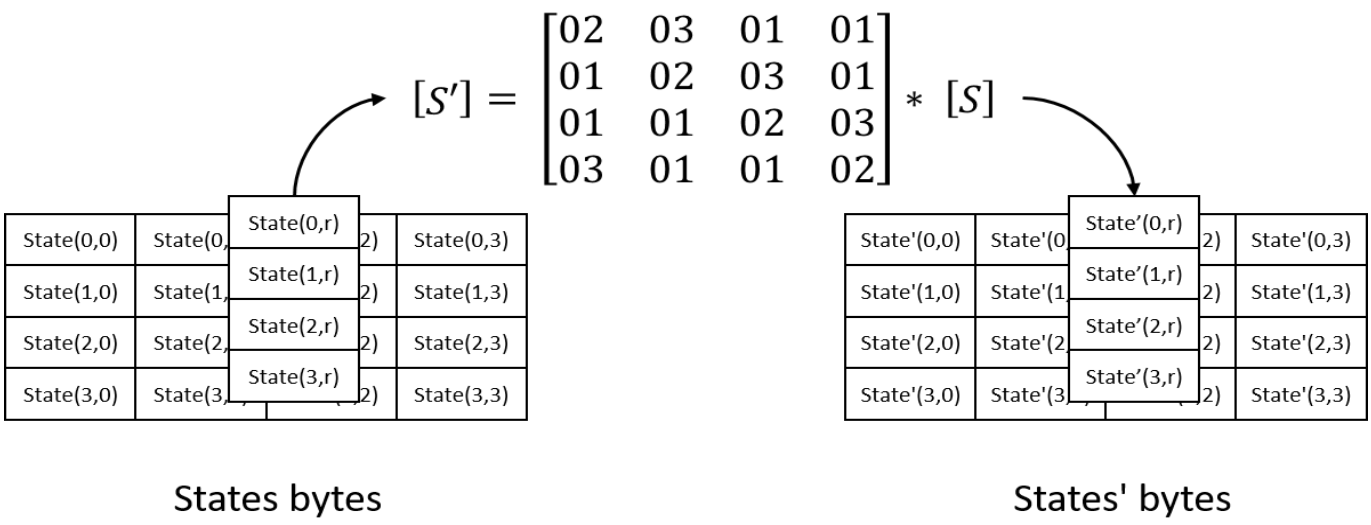


Fig. 6 Mix columns

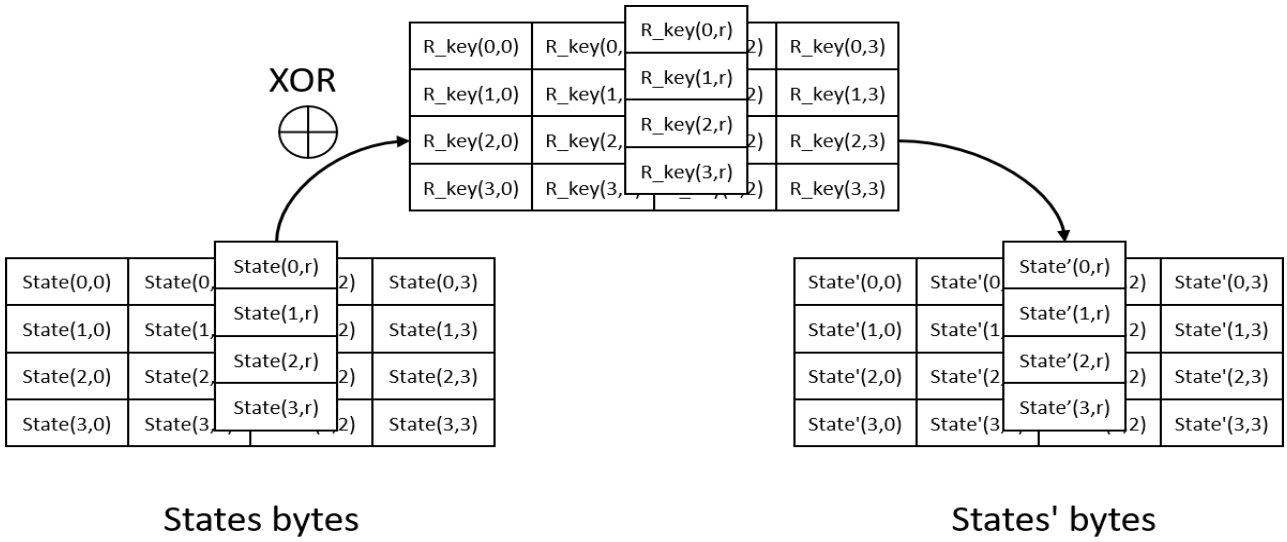


Fig. 7 Add round key

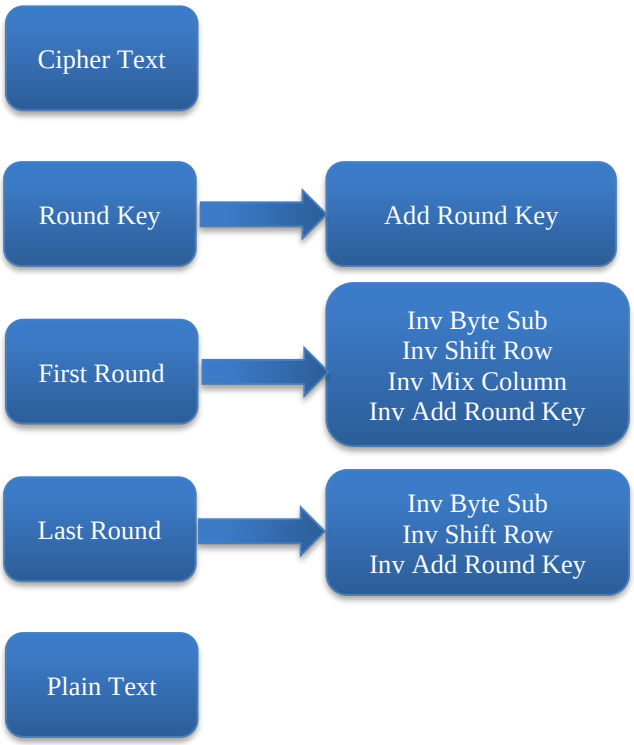


Fig. 8 Extensive Advanced Encryption Standard Algorithm is used for decryption

4.6. Decryption Process

Decryption converts cipher text back to ordinary text. EAES decryptions use previously produced file keys. Unlike standard AES decryptions, the processes are carried out in reverse order. Although these procedures are connected, they are implemented independently, as opposed to the Feistel Cipher. The suggested tag key scheme protects files, while deduplication eliminates data redundancy. SHA generates a unique fingerprint for files, which is subsequently encrypted by the proposed EAES alongside the file for maximum security. Thus, a user generates a key for each file saved in the cloud, and when he or she shares a file, the appropriate hash key is displayed. Another user can retrieve the file using the hash key provided by the user, but his or her access is confined to that file, keeping additional files safe or hidden from others. As a result, the number of blocks needed for encryption is greatly reduced.

This increases the amount of encryption. With the higher rate of encryption, the suggested EAES technique encrypts faster than AES. Table 2 shows a comparison of the proposed methodology to alternative redundancy reduction methods such as Data Level Redundancy Elimination (DRE), Role Based Access Control (RBAC), End-To-End (E2E), Content Services (CS), and Two-Factor Authentication (2FA). It takes place in the cloud in terms of processing time, performance, and storage space gain. The same is seen in Table 1 and Figures 9 and 10, respectively.

5. Results and Discussions

The proposed EAES method is defined as a symmetrical cipher block. During the Shift Rows transformation phase, the state's rows are regularly shifted at different offsets. Row 0 remains in position, while rows 1, 2, and 3 move one, two, and three bytes to the left. The Rijndael algorithm is used in this process of encryption. The key advantages of the improved AES algorithm are its very simple nature and fast processing speed without memory sacrifice. Another big advantage is the same hash key is used for both encryption and decryption. The Rounding function is used in AES encryption. The operation involving sub-bytes is a non-linear type of substitution on each byte using the specified table.

Step : 1 The Shift-row operation enables a circular motion of states.
Step : 2 The mix-column procedure includes multiplying bytes using a polynomial modulo.
Step : 3 The Add Round Key function adds a round key to the existing state.

The graph contrasts AES encryption with Improved AES encryption speeds. Because the size of each block in the extended AES algorithm surpasses that of the AES algorithm, the total number of blocks decreases. A Cloud Sim simulator implementation result revealed that the suggested modified AES was more efficient and effective than previous techniques, with data sizes ranging from 400,550,800,1020 and 1200 kilobytes. Figures 9 and 10 depict the results of a runtime comparison of AES and Extensive AES for various sizes.

$$\text{Runtime} = \text{the amount of data / pace} \quad (1)$$

Table. 2. Comparative Analysis of processing time in ms, Storage Attain and Efficiency

DRE	22.21	17.17	7.85	5.14
RBAC	18.85	16.24	6.76	5.19
E2E	25.56	21.18	5.87	3.20
CS	18.16	15.27	4.91	3.21
2FA	19.12	22.13	3.89	4.23
EAES	12.56	10.41	2.95	2.32

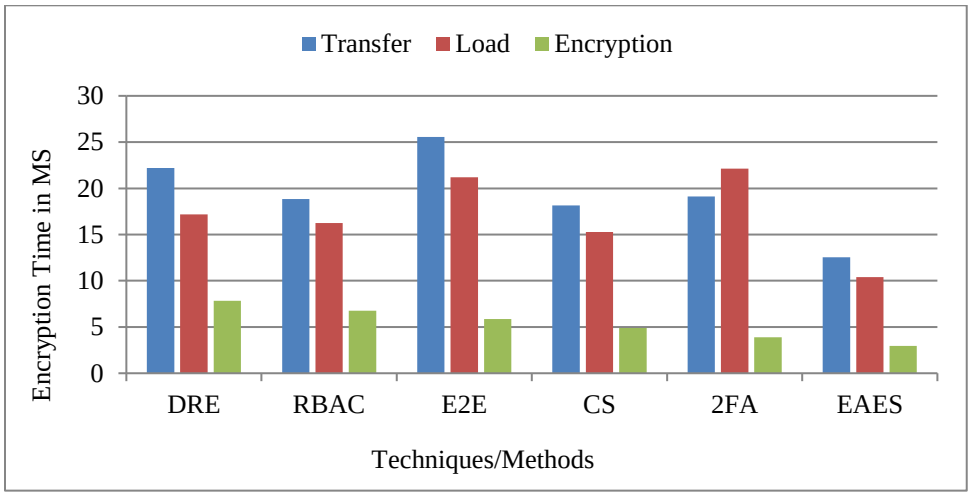


Fig. 9 Encryption processing time in MS

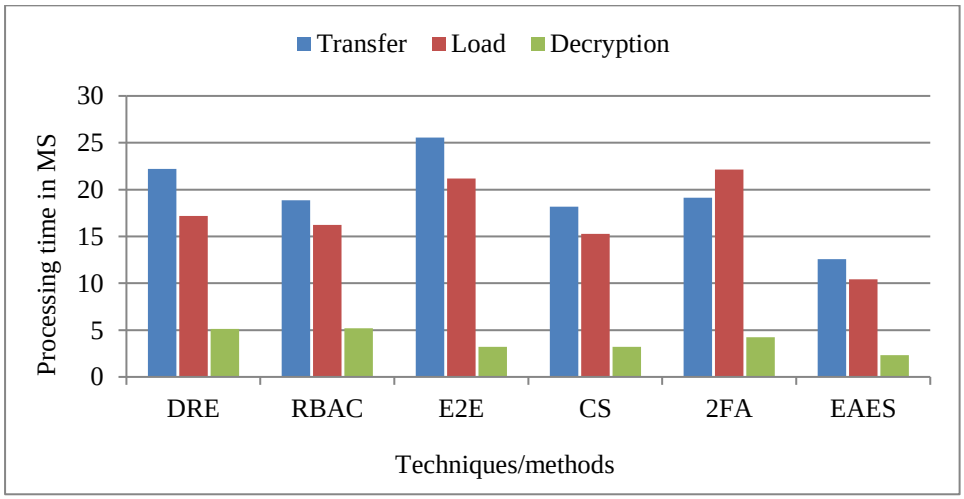


Fig. 10 Decryption Processing time in MS

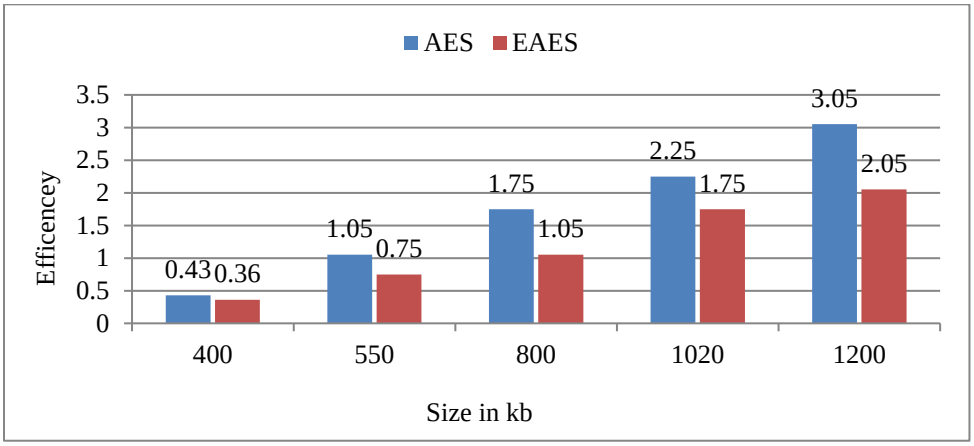


Fig. 11 Runtime comparison of AES and EAES

In general, the Avalanche Effect (Av) assesses algorithmic modifications. Simply said, even minor changes in input can result in significant differences in text output. To compute Av, divide the number of altered bits in the cipher bits by the total number of bits. Table 3 represents that the effect (Av) metric assesses how an algorithm changes. It basically states that a slight change in input can result in a significant change in consequence. The Avalanche Effect of our recommended technique and the underlying article were calculated using the following formula: 50 for key exchange and 80 for plain text. We can see that the bigger the avalanche effect, as in our suggested algorithm, the more difficult it is to interrupt the process. As a result, the Avalanche effect makes our method more secure. The previous table is visually depicted below, illustrating that our proposed solution is more secure than other encryption schemes.

AV equals the number of altered bits divided by the total number of bits (2)

Table. 3. Avalanche effect comparison

Techniques/Methods	1-bit Plain Text Change	Avalanche Effect	1-bit Key Change	Avalanche Effect
DRE	69	0.86	45	0.86
RBAC	54	0.65	38	0.76
E2E	71	0.88	64	1.28
2FA	82	1.02	65	1.31
EAES	**89**	**1.11**	**75**	**1.51**

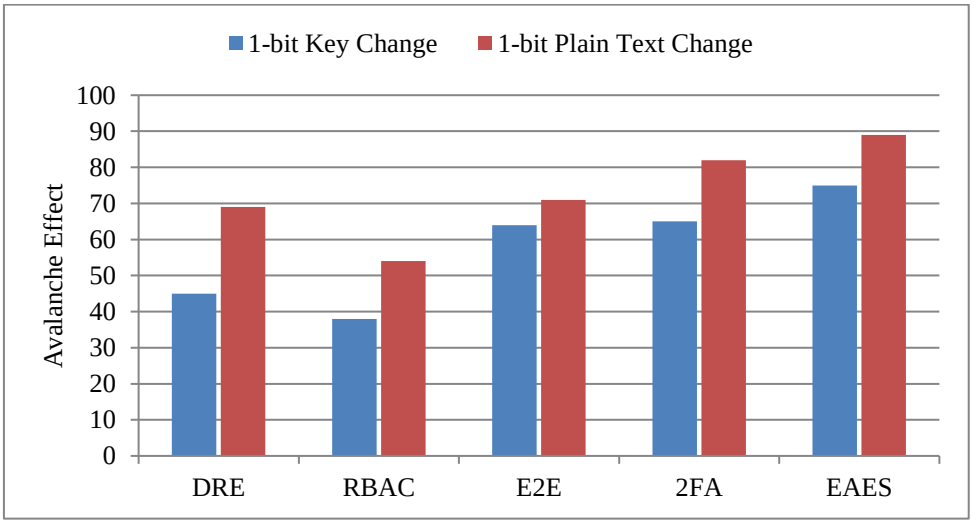


Fig. 12 Avalanche rffect

6. Conclusion

Security needs and improvements have been the major research area in cloud computing, particularly with the data, as most organizations and businesses are data-driven. Ownership, availability, access control mechanisms and privacy are the chief concerns as far as data security is concerned. Data protection in the cloud environment is much similar yet complex when compared to the traditional data center approaches. These include more complex and stronger mechanisms to provide authentication, encryption and authorizations for data. Based on the reports from investigations, security analysis and privacy concerns of the organizations, the need for adequate techniques to address these issues is the need for cloud computing. These are the primary reasons why firms are hesitant to shift their business/applications to the cloud. More efforts are required to secure the confidentiality, safety, and privacy of data in the cloud. The algorithm is the most crucial aspect when creating security solutions. In Step I of the study, the suggested system employs tag key-based data deduplication techniques to reduce data redundancy, save space, and, finally, improve storage efficiency. This device also employs the Improved AES standard data encryption technology, which enhances overall security. Finally, this employs the hash key approach in data encryption using SHA, which is produced dynamically in a certain manner to ensure system security. Experimental results reveal that the suggested method outperforms other current methods across a wide range of performance criteria. This report has described the experimental results obtained during the testing phases. The cloud simulator was used to run the tests with the bare minimum of device settings. The experimental data were then presented in the form of graphs for easier comprehension. The proposed methods proved to be more successful at various stages than the other methods.

References

[1] A.E. Adeniyi et al., "Implementation of a Block Cipher Algorithm for Medical Information Security on Cloud Environment: Using Modified Advanced Encryption Standard Approach," *Multimedia Tools and Applications*, vol. 82, pp. 20537-20551, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[2] Bijeta Seth et al., "Integrating Encryption Techniques for Secure Data Storage in the Cloud," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 4, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[3] Hyo-jun Lee et al. "De-Identification and Privacy Issues on Bigdata Transformation," *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, Busan, Korea (South), pp. 514-519, 2020 [CrossRef] [Google Scholar] [Publisher Link]

[4] Fursan Thabit et al., "A New Lightweight Cryptographic Algorithm for Enhancing Data Security in Cloud Computing," *Global Transitions Proceedings*, vol. 2, no. 1, pp. 91-99, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[5] Arijit Ukil, Debasish Jana, and Ajanta De Sarkar, "A Security Framework in Cloud Computing Infrastructure," *International Journal of Network Security & Its Applications*, vol. 5, no. 5, pp. 1-14, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[6] Noha MM. AbdElnapi, Fatma A. Omara, and Nahla F. Omran, "A Hybrid Hashing Security Algorithm for Data Storage on Cloud Computing," *International Journal of Computer Science and Information Security*, vol. 14, no. 4, pp. 175-181, 2016. [Google Scholar] [Publisher Link]

[7] Simona Samardjiskam, and Danilo Gligoroski, "An Encryption Scheme based on Random Split of St-Gen Codes," *2016 IEEE International Symposium on Information Theory (ISIT)*, Barcelona, Spain, pp. 800-804, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[8] Dustin Moody, and Ray Perlner, "Vulnerabilities of "McEliece in the World of Escher"," *7th International Workshop Post-Quantum Cryptography*, Fukuoka, Japan, pp. 104-117, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[9] Lidong Chen et al., "*Report on Post-Quantum Cryptography*," National Institute of Standards and Technology, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[10] Colin Ting Si Xue, and Felicia Tiong Wee Xin, "Benefits and Challenges of the Adoption of Cloud Computing in Business," *International Journal on Cloud Computing: Services and Architecture*, vol. 6, no. 6, pp. 1-15, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[11] Namje Park, and Namhi Kang, "Mutual Authentication Scheme in Secure Internet of Things Technology for Comfortable Lifestyle," *Sensors*, vol. 16, no. 1, pp. 1-16, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[12] Nabeel Zanoon, "Toward Cloud Computing: Security and Performance," *International Journal on Cloud Computing: Services and Architecture*, vol. 5, no. 5, pp. 17-26, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[13] L. Arockiam, and S. Monikandan, "Data Security and Privacy in Cloud Storage using Hybrid Symmetric Encryption Algorithm," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 8, pp. 3064-3070, 2013. [Google Scholar] [Publisher Link]

[14] Vikas Goyal, and Chander Kant, "An Effective Hybrid Encryption Algorithm for Ensuring Cloud Data Security," *Big Data Analytics: Proceedings of CSI 2015*, Singapore, pp. 195-210, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[15] Joseph Henry Anajemba et al., "Improved Advanced Encryption Standard with a Privacy Database Structure for IoT Nodes," *2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT)*, Gwalior, India, pp. 201-206, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[16] Lin Teng et al., "A Modified Advanced Encryption Standard for Data Security," *International Journal of Network Security*, vol. 22, no. 1, pp. 112-117, 2020. [Google Scholar] [Publisher Link]