Received: February 20, 2024. Revised: April 26, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Preserving Confidential Data Using Improved Rivest-Shamir Adleman to Secure Multi-Cloud

Harish Naik Bheemanaik Manjyanaik[1]* **Rajanikanta Mohanty[1]**
Jayanthi Mangayarkarasi Kannan[1]

[1]*Department of Computer Science and Engineering, Jain Deemed to be University, Bengaluru, India*
* Corresponding author's Email: harishnaikbm012@gmail.com

Abstract: Multi-cloud security is a comprehensive cloud security solution that protects applications and data through various platforms. Multi-cloud security contains public and private clouds like Amazon Web Service (AWS), Google Cloud Platform, Azure, and the infrastructure of Oracle Cloud. However, it is difficult to protect data in the environment of multi-cloud, and it is challenging to secure and track data stored through various cloud providers. In this research, the Improved Rivest-Shamir Adleman (IRSA) approach is proposed for multi-cloud security to protect confidential data. The user-owned data files are initially assumed to vary in size, typically falling within the range of 100 to 1000 MB. AWS S3 is utilized to allow users to retrieve and store any amount of data at any time. Then, the AWS Identity and Access Management (AWS-IAM) is employed to make secure management of access to the AWS resources for verification. IRSA is designed for encryption to prevent the decomposition of the prime factor from the public key. After the successful encryption of the various file parts, each file part is uploaded separately to a distant cloud using a file naming scheme. When compared to the existing methods like Enhanced Symmetric Key Encryption Algorithm (ESKEA), Homomorphic Bloom Filter-based Data Security (HBDaSeC), Triple Data Encryption Standard (TDES), Efficient Ciphertext-Policy Attribute Based Encryption (E-CP-ABE), hybrid cryptographic technique, and Advanced Encryption Standard (AES), the proposed IRSA achieves better encryption and decryption times of 103 ms and 110 ms in 100 data size respectively.

Keywords: Amazon web service, Google cloud platform, Identity and access management, Improved rivest-shamir adleman, Multi-cloud.

1. Introduction

Cloud technology is one of the primary developing areas in the Information Technology (IT) field. The cloud has various benefits such as flexibility, scalability, reliability, feasible collaboration, and unlimited storage [1]. One of the primary security problems in the cloud is protecting sensitive data [2, 3]. To address these problems, a Multi-Cloud (MC) environment is established to secure the data [4]. The MC generates the interface of a single web for accessing the resources from the platform of a heterogeneous cloud. The MC establishes the data sharing by the owner of data in the cloud [5]. The connectivity of the internet has permitted users to employ scalable distribution through the heterogeneous big data environment [6]. Some huge companies and organizations guarantee storage service performance and reliability but do not take risks with their client-sensitive data [7]. The cloud allows companies to employ shared resources as their own to establish and manage the infrastructure of computing. The various kinds of resources are shared like storage, Virtual Machine (VM/ container), or an application [8]. In cloud storage, a secure data deduplication technique is utilized to minimize the storage space when data copies are eradicated [9, 10]. Cyber-Physical Systems (CPS) are resource-constrained, tightly interconnected, large-scale dispersed cyber collection and physical-system elements [11].

The recent development in cloud-based Identity Management Systems (IMS) is the advanced version

DOI: 10.22266/ijies2024.0831.13

of traditional IMS that adopts large innovative technology such as user's security and signature model for assisting the cloud to validate the legitimacy of user's [12]. In a cloud, the multi-owner scheme is more robust than a single-user cloud scheme. In conventional public key cryptography and Attribute-Based Encryption (ABE), the ciphertext is not coded for a single user [13]. The ABE is divided into CP-ABE and Key-Policy ABE (KP-ABE) [14]. The cloud server performance is increased by employing various factors like data deduplication, load balancing, and task scheduling. Security is significant when the resource is employed in a private cloud. The cloud security execution is established by developing access control, authentication approach, integrity techniques, and applying confidentiality [15]. A mathematical approach for managing authentication like data integration and encryption is called cryptology [16]. In the process of deduplication, various approaches are employed to access the efficient approach. The existing RSA technique is the first secure public key cryptosystem for the transmission of data. The encryption time is greater in the RSA approach and it cannot be applied to the environment of single-user [17]. However, it is difficult to protect data in multi-cloud environment, and it is challenging to secure and track data stored through various cloud providers. The main contribution of this research is:

- AWS S3 is utilized to allow users to retrieve and store any amount of data at any time. It supports both Client-Side Encryption (CSE) and Server-Side Encryption (SSE) for Amazon S3 to protect the data at transit and rest upon unauthorized and unauthentic access when assuring available and interactive data.
- The AWS-IAM is used to make secure management of access to the AWS resources for verification.
- IRSA is designed for encryption to prevent the decomposition of the prime factor from the public key.

This research paper is given as follows: Section 2 determines the literature survey. The block diagram of the proposed technique is discussed in Section 3. The results are illustrated in Section 4. Section 5 discussed the conclusion.

2. Literature survey

The related works of this research are discussed along with their advantages and disadvantages.

Silambarasan Elkana Ebinazer [18] implemented an ESKEA to enhance data confidentiality. The data block-level deduplication was established by employing Convergent Encryption (CE) for checking the Cloud Service Provider (CSP) duplicate data copies. Then, the implemented ESKEA technique was employed to secure the storage of data. The selection of an Optimal Secret Key (OSK) increases the efficiency of the implemented ESKEA approach. However, during the execution of symmetric key encryption, the encryption and time complexity need to be minimized.

Bijeta Seth [19] presented a HBDaSeC technique to secure a storage of data in cloud. The system contains dual encryption and fragmentation of data which secure data distribution in a multi-cloud. Data storage in multi-cloud removes essential problem of vendor-lock in connected with single cloud. This approach enhances the memory by utilizing Routing Low Power and Lossy Network (RPL) which minimizes the latency and maximizes the throughput. However, handling and updating Bloom filters was intricate and resource-intensive which leads to bottleneck performance.

Mohan Naik Ramachandra [20] introduced a TDES to generate security for big data in cloud environment. The input was encrypted after choosing the data by utilizing the TDES approach. The encrypted data was then stored in the environments of the cloud which supports the data's read-write operation. Finally, the data decryption was performed to retrieve the data by employing TDES. This technique provides a simpler approach by maximizing the key size in the Data Encryption Standard (DES) to secure against attacks and receive data privacy. However, the TDES required to protect data due to its encryption process.

Yongkai Fan [21] suggested an E-CP-ABE to protect the data confidentiality. A blockchain was constructed for tracking data namely TraceChain which generates both users and data owner to track an outsourced data. The suggested approach was effective and practical by employing blockchain. However, E-CP-ABE has difficulties in handling attribute policies and sets which leads to scalability problems in large-scale system.

Manreet Sohal [22] developed a hybrid cryptographic technique in multi-cloud that employs Identity-based Broadcast Encryption (IBBE) to manage a symmetric key approach (BDNA). The primary aim of this approach was to prevent the data of users from untrusted CSP. To achieve this aim, the security approach divides the data of users into various parts and encrypts these parts by utilizing BDNA. Then, this approach generates an effective technique for securing the secret key produced by BDNA. This approach was applied in all applications that utilize the cloud to store their private data.

Received: February 20, 2024. Revised: April 26, 2024.

However, because of the availability of access tokens from clouds, maximum cloud limit was set to five in this approach.

Md. Alamgir Hossain [23] implemented a hybrid verification approach depending on biometrics and encryption system to enhance a security in cloud data. A unique finger impression was employed has biometric test and encryption has Advanced Standard Encryption (AES) computatioQn. One-Time Password (OTP) was performed as a secret key to increase security system. By employing this approach, the security was increased and generate user valid authentication. However, combining various verification layer maximize system complexity which result in usability problems and increased overhead.

3. Proposed methodology

The IRSA technique is proposed for multi-cloud security to protect data confidentiality. Initially, the user owned the data's file size of 100 to 1000 MB. To upload a file to the cloud, a user should log into the appropriate cloud storage accounts by utilizing valid credentials. To store data, verification is needed from the CSP which is assigned normally in a single cloud and split sensitively and stored in the various cloud. Once the verification is completed, the data is encrypted using IRSA. After a successful encryption of various parts of a file, each file part is uploaded to a distant cloud. Fig. 1 indicates a block diagram for proposed technique.

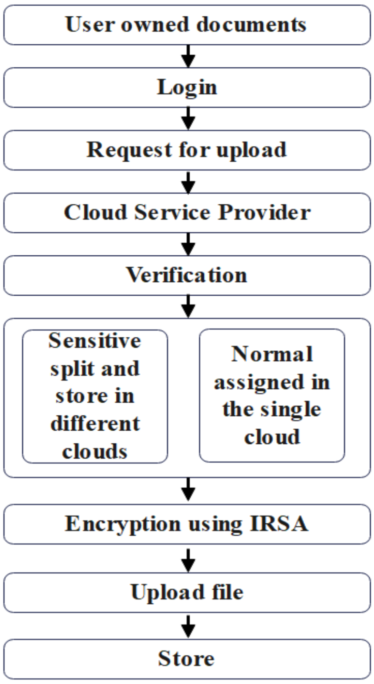

Figure. 1 Block diagram for proposed technique

3.1 User owned documents

Initially, the user-owned data files are considered to range in size from 100 to 1000 MB. To upload these files to the cloud, a user should log into the appropriate cloud storage accounts by utilizing valid credentials. By employing a secret key, the user uploads the data to the cloud. The Data Owner (DO) shares the file name and secret key upon receiving a request from the receiver. The DO's primary task is to manage an authorized list of users on the secret keys. If the user wishes to admittance the data, they are required to place a request for DO. It accelerates the secret key on the file name to the requested user via the secured channel. The requested user acknowledges the name of a file to an environment of multi-cloud and recaptures parts of encrypted data from multi-cloud. The requested data is fed into the procedure of CSP.

3.2 Cloud service provider using AWS S3

CSP is a third-party company that contributes scalable computing resources that can be accessed over a network, including cloud-based platforms, services or applications, and storage. Amazon Web Service S3 (AWS S3) is a scalable storage object service generated by AWS that enables developers and businesses to retrieve and store data on the Internet. It supports both client-side and SSE for Amazon S3 to protect the data at transit and rest upon unauthorized and unauthentic access when assuring available and interactive data. The data is notable as unallocated from the storage block to delete the data but not the basic physical media. While new data is composed of the storage of blocks, the data could be zeroed out. In AWS S3, huge files are divided into smaller chunks using the Amazon S3 Multipart upload feature. These chunks are then stored in the S3 storage buckets which generate an effective and scalable way to manage huge files where each part is assigned a unique identifier and denotes collectively the original file.

3.2.1. Protect data in transit

CSE is utilized for protecting the data in transit by encrypting data before transmitting it to AWS S3. The Hypertext Transfer Protocol Secure (HTTPS) is employed for secure connection. The following approaches are provided to secure/ encrypt among its source or destination and Amazon S3.

- To establish confidential access to data, the Evident Security Platform (ESP) Secure Sockets Layer (SSL), or Internet Protocol Security (IPSec) is used for data encryption in transit.

- The data integrity is authenticated by utilizing the Authentication Header (AH), IPSec ESP, or SSL/TLS.
- To establish the accurate authorized and authenticated user, the connection of SSL/TLS is employed on the authentication of the server certificate depending on server's alternative name or common name.

3.2.2. Protecting data at rest

The unique encryption key is provided for each object in the SSE. Then, by employing Advanced Encryption Standard-256 bits (AES-256), the data is encrypted. Finally, the regularly rotated and securely stored master key is encrypted by the encryption key itself. The users are selected among the possibilities of mutual exclusive to handle the keys of encryption:

- Users will utilize Amazon S3-Managed Keys by encryption of a strong multifactor. By employing AES-256 encryption, the data will be encrypted then the key of encryption through a master key.
- The user employs the AWS Key Management System (SSE-KMS). For unauthorized access, SSE-KMS encryption includes SSE-S3 encryption by using separate permission for employing the key of encryption data for protecting data.
- The Customer-Provided Keys (SSE-C) is applied by a user. The user is reliable for providing and handling encryption keys when Amazon S3 handles encryption and decryption procedures.

However, AWS generates two encryption possibilities for client-side encryption. First, one is the responsibility of the user, in this approach the users are responsible for establishing and handling encryption keys. In the plane text, the keys will not be supplied to AWS only the application of the user manages the data encryption before submission to Amazon S3. After admitting it from Amazon S3, users are also managing the data decryption. The second one is AWS-KMS. The user can generate an AWS-KMS in this approach to their ID of master key and request for the encryption of data. Additionally, the AWS handles the customer's encrypt key and master key to preserve integrity.

3.3 Verification using AWS-IAM

For storing the data, verification is generated from the CSP which is assigned normally in a single cloud and split sensitively and stored in the various cloud. If the normal file is designed as non-sensitive, it will be controlled by a single Virtual Machine (VM). When a file is sensitive, it will be separated into different pieces depending on data size. A random VM is allotted to every part of the data, it can handle a large number of files at once. AWS-IAM is a service that makes secure management of access to the resources of AWS for verification. It enables users to control who can access particular resources and actions performed by them. IAM generates a centralized system for both authorization and authentication within the cloud of AWS which increases security. IAM has a policy conception which is the representation of high-level actions a user is permitted to execute on resources. The IAM service in AWS enables users or businesses to assist in fine-grained access control among different resources and services. Moreover, developers cannot have access rights. The AWS's owner account has all the authority from the beginning and establishes the initial permission that provides other developers to work in similar areas.

To establish this, the IAM approach creates various types of IAM resources like User, Role, User Group, AWS resources, Tag, and so on. The resources of IAM are not intended for regular AWS resources; AWS resources like compute databases or clusters are services where IAM resources apply to data components employed to access these AWS resources. Moreover, the resources of IAM are matched and associated with various access control approaches like Discretionary Access Control (DAC), Role-Based Access Control (RBAC), Mandatory Access Control (MAC), and Attribute-based Access Control (ABAC). These policies are the IAM's primary building blocks enabling scalability, security, and manageability. Also, AWS employs Amazon Resource Names (ARNs) assigned to every component to identify unique resources across the whole AWS platform. Each policy is established by utilizing JSON files with various parameters based on the policy type. The primary parameters are Effect, Action, Principal, Statement, Condition, and Resources. In addition, there are certain semi-operational parameters like Statement ID (Sid) and Version which are employed for metadata and version control respectively. After verification, the data is encrypted using IRSA.

3.4 Encryption algorithm

Once the verification is completed, the data is encrypted using IRSA. The RSA is a well-established public-key cryptosystem for digital signature and encryption. In a limited time, one can able to reversely factor the multiplication of two large prime numbers. This means that one can efficiently crack the RSA encryption approach. Here, IRSA is designed to prevent the decomposition of the prime factor from the public key. Initially, the master-slave chaotic synchronization system is employed to acquire identical random signals in the transmitter and receiver. Corresponding to these synchronized random signals, then the prime pair and public keys are randomly generated in the transmitter and receiver simultaneously. Hence, the probability of prime factor decomposition is removed.

3.4.1. Rivest-shamir-adleman (RSA)

RSA asymmetric encryption approach [24] is primarily employed for data encryption and decryption with various private and public keys. The public key is employed in the encryption process and the private key is utilized in the decryption process. The public key is open and the private key is maintained by individuals. Here, the public key is further tuned into an invisible public key or dynamic private key depending on the synchronization of chaos dynamics. The architecture of RSA is represented in Fig. 2.

The mathematical formula for the traditional RSA encryption approach is expressed in Eqs. (1) to (6).

$$\textbf{Step 1:}\text{ Define } N = p \times q \qquad (1)$$

Where N represents numbers, p and q are the two primes that are selected randomly.

Step 2: The Euler function $\varphi(N)$ is defined which is expressed in Eq. (2).

$$\varphi(N) = (p-1) \times (q-1) \qquad (2)$$

Where $\varphi(N)$ is the Euler function

Step 3: Another number e is determined which is expressed in Eq. (3).

$$\gcd(\varphi(N), e) = 1 \qquad (3)$$

Step 4: The natural number d is established that is expressed in Eq. (4).

$$(d \times e)(mod\ \varphi(N))) = 1 \qquad (4)$$

Step 5: The pair (N, e) is a public key and (N, d) is the private key which is maintained by individuals. Then, after obtaining the (N, e) and (N, d), the encryption and decryption approach is expressed in Eqs. (5) and (6).

$$E(M, e, N) = M^e(mod\ N) = C \qquad (5)$$

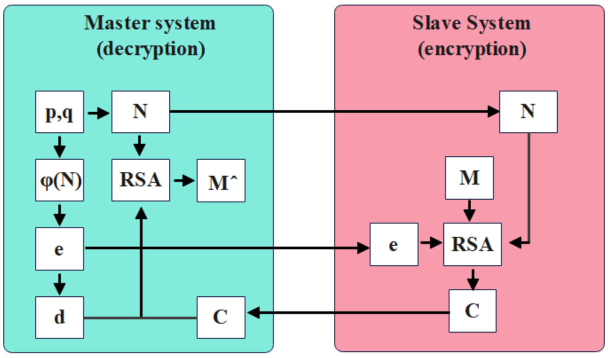

Figure. 2 Architecture of RSA

$$D(C, d, N) = C^d(mod\ N) = \widehat{M} \qquad (6)$$

Where M is the original signal and C is the encrypted signal, $E(M, e, N)$ is the function of encryption, $\widehat{M}$ is the recovered signal which is performed by employing the decryption approach $D(C, d, N)$.

3.4.2. Improved RSA

The RSA encryption algorithm security depends on the difficulty of reversing the decomposition of the prime factor for the large number $N = p \times q$. In a limited time, one can able to reversely factor the multiplication of two large prime numbers. This means that one can efficiently crack the RSA encryption approach by Shor's algorithm with the quantum computer of high speed. Hence, IRSA approach is established which will hide the public key to prevent it from being cracked by the Shor approach. In public channels, the opening public key step is eliminated to enable the factorization probability towards the (N, e) public key that is decreased to zero. The (N, e) public key production in IRSA is generated by the master-slave chaotic synchronization system. Fig. 3 shows the architecture of IRSA.

Initially, the chaotic system of master-slave occurs by the synchronization controller. The synchronization signal of random chaos is acquired at the same time in the transmitter and receiver for generating the identical random prime numbers $p \times q$ and other corresponding parameters depending on the traditional RSA approach. Therefore, the (N, e) public key cannot be required to arrive in a public channel and greater security is attained. By employing chaotic synchronization, the identical random signal can be acquired in the master-slave chaotic system. However, the RSA approach needs two prime numbers $p \times q$ and e as the decryption and encryption parameters. However, the chaotic

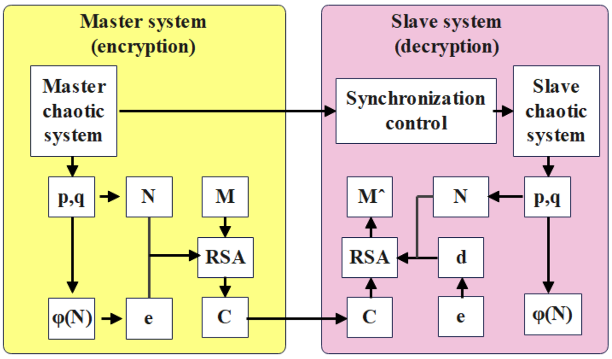

Figure. 3 The architecture of IRSA

random signal does not enable them to be primed. Hence, the identical prime number array is established in the transmitter and receiver that maps a pair of primes according to the signals of random chaos. The primes in both the transmitter and receiver are random and identical due to the synchronized random chaotic signals. Therefore, the improved algorithm is modified which is expressed in Eqs. (7) and (8).

$$E(M,e,m_p,m_q) = M^e\left(mod(m_p \times m_q)\right) = C \quad (7)$$

$$D(C,d,s_p,s_q) = C^d\left(mod(s_p \times s_q)\right) = \widehat{M} \quad (8)$$

Where m_p, s_p, m_q, s_q are the prime pairs by utilizing the prime approach. The $m_p = s_p$, $m_q = s_q$, and $M = \widehat{M}$ are ensured since the chaotic systems of master-slave are synchronized.

3.5 Upload file

After the successful encryption of various file parts, each file part is uploaded to a distant cloud. These parts are randomly established and feasible to one or more clouds that cannot obtain any part of an inclined file. Various users have a file with an identical name, hence when uploading a file with an identical name, there is a probability that a file with the same name may already exist in the cloud, leading to overwritten on that file. To address these problems, a file naming approach is employed where the name of the file is added with the unique value of timestamp, hence cannot have two file names on the same name. Finally, the file name will be stored in the cloud effectively. Notation Description is given in Table 1.

Table 1. Notation Description

Symbol	Description
N	Numbers
p and q	Two primes
$\varphi(N)$	Euler function
d	Natural number
(N,e)	Public key
(N,d)	Private key
M	Original signal
C	Encrypted signal
$E(M,e,N)$	Encryption function
$\widehat{M}$	Recovered signal
$D(C,d,N)$	Decryption function
m_p, s_p, m_q, s_q	Prime pairs
e	Encryption and decryption parameters

4. Experimental results

The IRSA is simulated by employing a Python 3.8 environment with RAM:16GB, Processor: Intel core i5, Operating System: Windows 10, and GPU: 6 GB. The parameters like encryption time (ms) and decryption time (ms), Packet Delivery Ratio % (PDR), and Latency (ms) are utilized to estimate the performance of the model.

4.1 Qualitative and quantitative analysis

Here, IRSA approach is performed by concerning encryption, decryption time (ms), PDR, and latency are presented in Tables 2 to 4. Table 2 indicates performance analysis of the data size vs encryption time. The performance of the AES and RSA is compared with the proposed IRSA approach. Fig. 4 represents the graphical representation of data size vs encryption time. The acquired outcomes represent that IRSA attains better encryption time of 103 (ms) in 100 (MB) data size respectively.

Table 3 illustrates the performance analysis of the data size vs decryption time. The performance of the AES and RSA is compared with the proposed IRSA approach.

Table 2. Performance analysis of data size vs encryption time

Data Size (MB)	Encryption time (ms)		
	AES	RSA	IRSA
100	310	280	103
200	370	367	110
300	419	406	186
400	490	482	293
500	570	565	343
600	615	606	366
700	799	788	485
800	894	826	639
900	900	863	782
1000	896	851	804

Received: February 20, 2024. Revised: April 26, 2024.

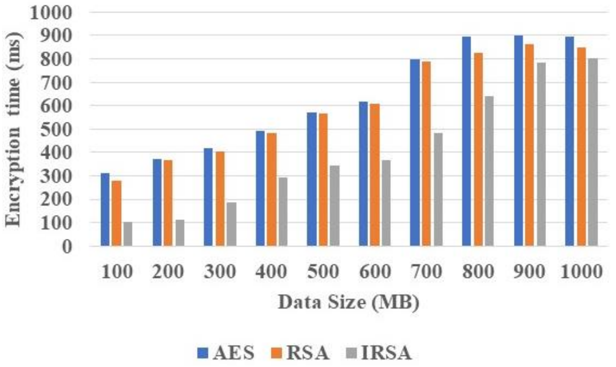

Figure. 4 Graphical representation of data size vs encryption time

Table 4. Performance analysis of the data size vs execution time

Data size (in MB)	Execution time (in ms)		
	AES	RSA	IRSA
100	350	316	239
200	369	354	278
300	392	378	295
400	419	396	360
500	439	420	411
600	465	459	446
700	554	527	514
800	597	586	582
900	673	669	653
1000	691	675	676

Fig. 5 represents the graphical representation of data size vs decryption time. The acquired outcomes illustrate that IRSA attains a better decryption time of 110 (ms) in 100 (MB) data size respectively.

Table 4 illustrates the performance analysis of the data size vs execution time for the proposed IRSA. The time taken for transmission of encryption and decryption in data size is the execution time in the IRSA approach.

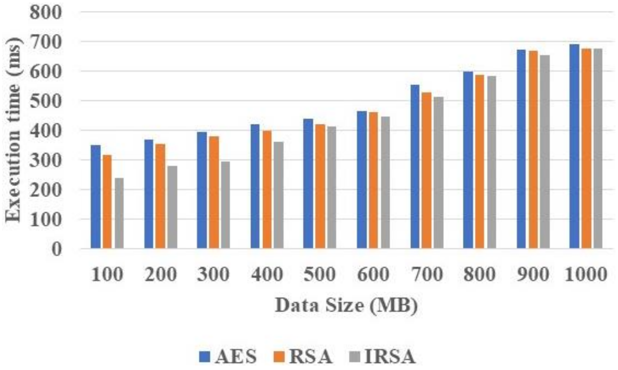

Figure. 6 Graphical representation of data size vs execution time

The performance of the AES and RSA is compared with the proposed IRSA approach for execution time. Fig. 6 represents the graphical representation of data size vs execution time. The proposed IRSA achieves 239 ms execution time in 100 MB data size compared to existing approaches.

4.2 Comparative analysis

The comparative analysis of IRSA with existing methods are presented in Table 5. The existing methods, ESKEA [18], HBDaSeC [19], TDES [20], E-CP-ABE [21], Hybrid cryptographic [22], and ABE employed to evaluate the IRSA approach. The encryption time, decryption time, PDR, and latency values for [19, 21, 22, and AES are simulated based on the proposed approach scenarios. Fig. 7 represents the graphical representation of comparative analysis with existing methods with encryption (ms), decryption time (ms), and latency (ms). When compared to the ESKEA, HBDaSeC, TDES, E-CP-ABE, Hybrid cryptographic, and ABE, the proposed IRSA achieves a better encryption time of 103 ms, a decryption time of 110 ms, 96% of PDR, and latency of 12 ms in 100 MB datasize respectively.

Table 5. Comparative analysis with existing methods

Methods	Data size (MB)	Encryption time (ms)	Decryption time (ms)	PDR (%)	Latency (ms)
ESKEA [18]	100	125	150	N/A	N/A
HBDaSeC [19]		150	185	88	20
TDES [20]		204	428	N/A	14
E-CP-ABE [21]		210	354	91	18
Hybrid cryptographic [22]		245	390	93	16
AES		310	230	94	14
Proposed IRSA		103	110	96	12

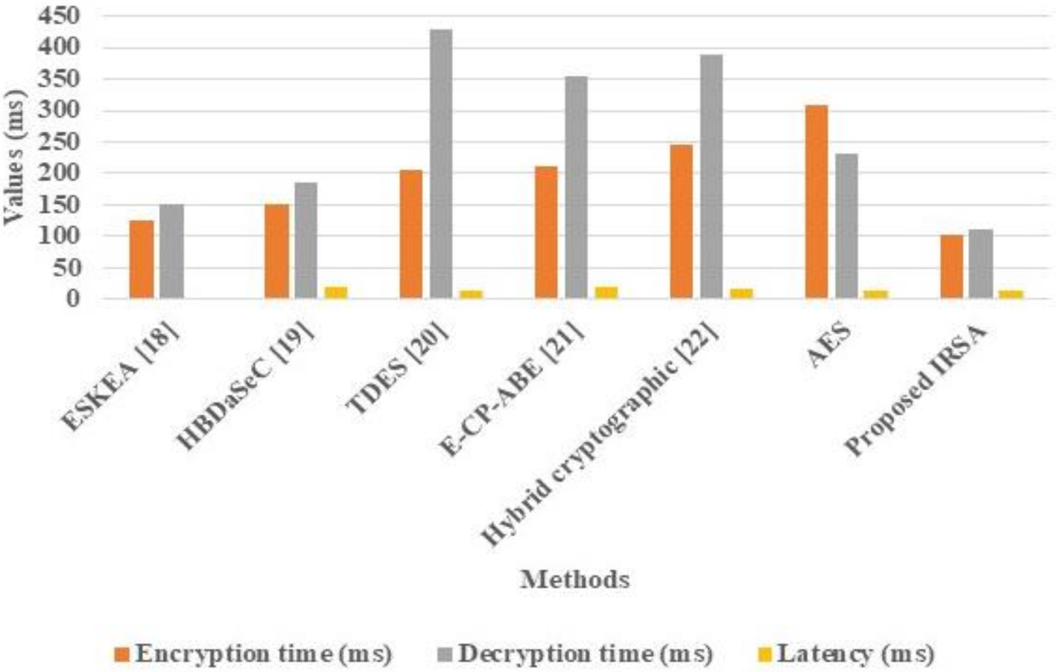

Figure. 7 Graphical representations of comparative analysis with existing methods with encryption and decryption time (ms), and latency (ms)

4.3 Discussion

Here, advantages of IRSA and limitations of existing techniques are discussed. The existing approaches have certain limitations like ESKEA [18] the encryption and time complexity need to be minimized during the execution of symmetric key encryption. HBDaSeC [19] handling and updating Bloom filters was intricate and resource-intensive which leads to bottleneck performance. TDES [20] required to protect data due to its encryption process. E-CP-ABE [21] has difficulties in handling attribute policies and sets which leads to scalability problems in large-scale system. The proposed IRSA model overcomes the existing model limitations. AWS S3 provides unlimited capacity of storage, reliability, scalability, and durability through redundant storage. The IRSA improves the data confidentiality in the environments of multi-cloud by generating stronger encryption which minimizes the unauthorized risks. When compared to existing techniques like ESKEA, HBDaSeC, TDES, E-CP-ABE, Hybrid cryptographic, and ABE, the proposed IRSA achieves better encryption and decryption times of 103 ms and 110 ms, in 100 data sizes respectively.

5. Conclusion

The IRSA technique is proposed for multi-cloud security to protect confidential data. Initially, the user-employed data files ranged in size from 100 to 1000 MB. To upload these files to the cloud, a user should log into the appropriate cloud storage accounts by utilizing valid credentials. AWS S3 is utilized to allow users to retrieve and store any amount of data at any time. Then, the AWS-IAM is used to make secure management of access to the AWS resources for verification. IRSA is designed for encryption to prevent the decomposition of the prime factor from the public key. The proposed IRSA achieves better encryption and decryption times of 103 ms and 110 ms, in 100 data sizes compared to existing approaches like ESKEA, HBDaSeC, TDES, E-CP-ABE, Hybrid cryptographic, and ABE. In the future, the analysis of CPU usage, computational time, and network utilization will be considered in the multi-cloud environment.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

The paper conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing—original draft preparation, writing—review and editing, visualization, have been done by 1st author. The supervision and project administration, have been done by 2nd author.

References

[1] U.R. Saxena, and T. Alam, “Role-based access using partial homomorphic encryption for securing cloud data”, *International Journal of System Assurance Engineering and Management*, Vol. 14, No. 3, pp. 950-966, 2023.

[2] D. Shivaramakrishna, and M. Nagaratna, “A novel hybrid cryptographic framework for secure data storage in cloud computing: Integrating AES-OTP and RSA with adaptive key management and Time-Limited access control”, *Alexandria Engineering Journal*, Vol. 84, pp. 275-284, 2023.

[3] U.R. Saxena, and T. Alam, “Role based access control using identity and broadcast based encryption for securing cloud data”, *Journal of Computer Virology and Hacking Techniques*, Vol. 18, pp. 171-182, 2022.

[4] G. Viswanath, and P.V. Krishna, “Hybrid encryption framework for securing big data storage in multi-cloud environment”, *Evolutionary Intelligence*, Vol. 14, pp. 691-698, 2021.

[5] Y. Ming, B. He, and C. Wang, “Efficient revocable multi-authority attribute-based encryption for cloud storage”, *IEEE Access*, Vol. 9, pp. 42593-42603, 2021.

[6] K. Sharma, A. Agrawal, D. Pandey, R.A. Khan, and S.K. Dinkar, “RSA based encryption approach for preserving confidentiality of big data”, *Journal of King Saud University-Computer and Information Sciences*, Vol. 34, No. 5, pp. 2088-2097, 2022.

[7] T. Shahien, A.M. Sarhan, and M.A. Alshewimy, “Multi-server searchable data crypt: searchable data encryption scheme for secure distributed cloud storage”, *Journal of Ambient Intelligence and Humanized Computing*, Vol. 12, pp. 8663-8681, 2021.

[8] M.B. Qureshi, M.S. Qureshi, S. Tahir, A. Anwar, S. Hussain, M. Uddin, and C.L. Chen, “Encryption Techniques for Smart Systems Data Security Offloaded to the Cloud”, *Symmetry*, Vol. 14, No. 4, p. 695, 2022.

[9] M. Pavithra, M. Prakash, and V. Vennila, “BGNBA-OCO based privacy preserving attribute based access control with data duplication for secure storage in cloud”, *Journal of Cloud Computing*, Vol. 13, No. 1, p. 8, 2024.

[10] N. Mageshkumar, J. Swapna, A. Pandiaraj, R. Rajakumar, M. Krichen, and V. Ravi, “Hybrid cloud storage system with enhanced multilayer cryptosystem for secure deduplication in cloud”, *International Journal of Intelligent Networks*, Vol. 4, pp. 301-309, 2023.

[11] R. Priyadarshini, A. Quadir Md, N. Rajendran, V. Neelanarayanan, and H. Sabireen, “An enhanced encryption-based security framework in the CPS Cloud”, *Journal of Cloud Computing*, Vol. 11, No. 1, p. 64, 2022.

[12] S. Pachala, C. Rupa, and L. Sumalatha, “l-PEES-IMP: lightweight proxy re-encryption-based identity management protocol for enhancing privacy over multi-cloud environment”, *Automated Software Engineering*, Vol. 29, No. 1, p. 4, 2022.

[13] S. Raj, and B. Arunkumar, “Enhanced encryption for light weight data in a multi-cloud system”, *Distributed and Parallel Databases*, Vol. 41, No. 1-2, pp. 65-74, 2023.

[14] S. Raj, B.A. Kumar, and G.K.D. Venkatesan, “A security-attribute-based access control along with user revocation for shared data in multi-owner cloud system”, *Information Security Journal: A Global Perspective*, Vol. 30, No. 6, pp. 309-324, 2021.

[15] M. Saravana Karthikeyan, R. Sasikala, N. Karthikeyan, and S. Karthik, “Improved performance of cloud servers using LBSDD factors of private cloud”, *Journal of Ambient Intelligence and Humanized Computing*, Vol. 12, pp. 5825-5834, 2021.

[16] M. Suganya, and T. Sasipraba, “Stochastic Gradient Descent long short-term memory based secure encryption algorithm for cloud data storage and retrieval in cloud computing environment”, *Journal of Cloud Computing*, Vol. 12, p. 74, 2023.

[17] B. Rasina Begum, and P. Chitra, “ECC-CRT: an elliptical curve cryptographic encryption and Chinese remainder theorem based deduplication in cloud”, *Wireless Personal Communications*, Vol. 116, No. 3, pp. 1683-1702, 2021.

[18] S. Elkana Ebinazer, N. Savarimuthu, and S. Mary Saira Bhanu, “ESKEA: enhanced symmetric key encryption algorithm based secure data storage in cloud networks with data deduplication”, *Wireless Personal Communications*, Vol. 117, pp. 3309-3325, 2021.

[19] B. Seth, S. Dalal, V. Jaglan, D.N. Le, S. Mohan, and G. Srivastava, “Integrating encryption techniques for secure data storage in the cloud”, *Transactions on Emerging Telecommunications Technologies*, Vol. 33, No. 4, p.e4108, 2022.

[20] M.N. Ramachandra, M. Srinivasa Rao, W.C. Lai, B.D. Parameshachari, J. Ananda Babu, and K.L. Hemalatha, “An efficient and secure big

data storage in cloud environment by using triple data encryption standard", *Big Data and Cognitive Computing*, Vol. 6, No. 4, p. 101, 2022.

[21] Y. Fan, X. Lin, W. Liang, J. Wang, G. Tan, X. Lei, and L. Jing, "TraceChain: A blockchain-based scheme to protect data confidentiality and traceability", *Software: Practice and Experience*, Vol. 52, No. 1, pp.115-129, 2022.

[22] M. Sohal, S. Bharany, S. Sharma, M.S. Maashi, and M. Aljebreen, "A Hybrid Multi-Cloud Framework Using the IBBE Key Management System for Securing Data Storage", *Sustainability*, Vol. 14, No. 20, p. 13561, 2022.

[23] M.A. Hossain, and M.A. Al Hasan, "Improving cloud data security through hybrid verification technique based on biometrics and encryption system", *International Journal of Computers and Applications*, Vol. 44, No. 5, pp. 455-464, 2022.

[24] V.C. Osamor, and I.B. Edosomwan, "Employing scrambled alpha-numeric randomization and RSA algorithm to ensure enhanced encryption in electronic medical records", *Informatics in Medicine Unlocked*, Vol. 25, p. 100672, 2021.