Received: September 30, 2023. Revised: November 16, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

An Efficient and Secured Big Data Storage in a Cloud-based Environment Using Hybrid Cryptography Algorithm and Rivest, Shamir, Adleman Algorithm

Aruna Kumari Koppaka[1]* **Vadlamani Naga Lakshmi[1]**

[1]*GITAM (Deemed to be university), India*
* Corresponding author's Email: aruna.jeshwin@gmail.com

Abstract: In recent years, big data analysis has been one of the hot research topics which attracts more researchers to work on it. Big data is comprised of highly sensitive data which is needed to be secured and monitored effectively. However, problems related to privacy and security have been provoked due to ineffectiveness in storing the data and relying on third parties. To overcome the issues of storing the data in a cloud environment securely, this research introduced a hybrid cryptography algorithm that is comprised with advanced encryption standard (AES), triple data encryption standard (TDES), and blow fish algorithm (BFA). Initially, the data from the user are acquired and uploaded to the cloud storage space, then a key is generated which helps in encrypting and decrypting the user's data. Process of encryption and decryption takes place using the proposed hybrid cryptography algorithm which effectively securely stores the data and helps the user based on their needs. Moreover, the rivest, shamir, adleman (RSA) algorithm is used in the process of key generation which helps to encrypt and decrypt the messages. The efficiency of the proposed approach is compared with the existing triple data encryption standard (TDES), blow fish algorithm – spotted hyena optimization (BFA-SHO) algorithm and secure data deletion and verification (SDVC) scheme on based on ciphertext policy-attribute based encryption (CP-ABE). The results obtained through analysis exhibits that suggested methodology took 183450 ms for executing the file size of 20 MB whereas the existing BFA-SHO took 248107 ms for executing the same file size of 20 MB.

Keywords: Big data, Cloud storage, Cryptography, Decryption, Encryption, Hybrid algorithm.

1. Introduction

In recent years, the usage of distributed computing systems such as network, autonomic, cloud, and pervasive has reached its peak. Especially, cloud based computing networks are well known and used by more organizations and individuals to store data [1]. Cloud computing is a kind of computing network that is linked with the internet to share a decentralized service based on the user's needs [2, 3]. Cloud offers various services such as infrastructure, software, etc., but providing security to the data stored in the cloud is a challenging task [4]. Generally, the user may have stored some of the sensitive information in the cloud environment and does not have a clear idea about the secureness of the data. The cloud is comprised of different types of resources such as networks, operating systems and management of memory space [5, 6]. One of the key benefits of cloud computing is that it enables cross-sector data access, real-time calculation, and cross-jurisdictional decision-making using cloud resources [7, 8]. However, data exchange between nodes, users, or across the cloud platform is extremely insecure until a strong protective solution is offered.

In the age of cloud computing, security, and seamless information sharing of private data, particularly multimedia data, are major challenges. Enabling computational efficiency is equally important to ensuring safe communication since real-time applications impose fast and reliable processing. Technically, a cloud is a storage space for digitalized data spread across a number of servers that are overseen by cloud service providers (CSP) [9, 10]. Cloud storage is widely utilized to allow for the storage of huge amounts of data known as big-data [11]. According to their needs, advanced

organizations store massive amounts of data in the cloud, which is easily subject to security attacks [12]. As a result, the primary concern of this work is data security in the cloud. Several cryptographic derivations are identified for exploring the challenges in providing data security and assurance [13]. Data security is intended to be used as a means of encrypting messages and decrypting them at the client end. Enhanced security derivations are added to the fragile data that is outsourced on the cloud by successfully insuring and constructing advanced cryptographic procedures [14, 15]. The enhanced security for cloud system is introduced using hybrid cryptography algorithm which is much helpful to the store the data in a secured manner.

The significant presentations shown in the research are as follows:

(1) The research developed a hybrid cryptography algorithm by combining the goodness of AES, TDES and BFA to secure the user data.
(2) The speed and strong security is achieved using AES, the compatibility is achieved using 3DES and the blowfish algorithm is known for its efficiency. Though the combination of algorithm results in better security, the key management is a complex process so this research utilized RSA for an effective key management.
(3) To perform effective encryption of user data to store it in the secured cloud storage platform and decrypt the data file when the proper key is provided from the user end.

The rest of sections are structured as follows: Section 2 is about the related works and the suggested method is presented in section 3. Section 4 discusses outcome of suggested approach and section 5 describes the conclusion.

2. Related works

The existing research that utilized various methodologies to offer security to the data stored in the cloud platform are discussed in this section.

Ramachandra [16] have introduced a triple data encryption standard (TDES) method to confirm the secureness of big data in cloud-based environment. TDES technique was less complex which relatively enhanced the size of the keys present in the data encryption standard (DES) to safeguard the information and preserve the data's privacy. After the stage of selecting the data from the dataset, the encryption was processed using the TDES technique. However, the utilization of memory space and the network is higher.

Devmane [17] have introduced a ring character hash (RCH) with a ring elliptical curve cryptography (RECC) homomorphic module to minimize the processing time and improve the rate of trustworthiness. At the initial stage, the data bytes are trained to the system and transformed into the binary values to evaluate the hash value using RCH mechanism. RECC was utilized to encrypt the data using the homomorphic property and the verification was employed to verify the effectiveness of the data and the decryption was processed using the secret key. However, the brute force attack lacks in confidential rate while transmitting the data through the channels.

Sundar [18] have introduced an enhanced cloud security model using quantum key distribution protocol (ECSM-QKDP) to offer cloud storage security with data dynamics and quantum key cryptography. Additionally, the communication among the individuals like cloud server, data owner, and legitimate user (LU) was confirmed. In initial phase, BB84 QKDP was utilized and in the second phase secured authentication protocol was created to bind the distance and the secure keys. Moreover, the secured quantum keys were transmitted to the model via a trustworthy QKD channel to perform the authentication process. However, the ECSM-QKDP was not suited for large-level data.

Shrivastava [19] have introduced a blockchain-based modified infinite chaotic elliptic cryptography (MICEC) to enhance and tighten the security of the cloud server. The MICEC was comprised of three phases such as authenticity protection, protection of ownership, and validating the identity mapping. MICEC was used for the authentication process which integrates the cryptography of elliptic curve and chaotic neural network to generate key and encrypt the data. However, the data stored in the Blockchain using MICEC cannot be modified again.

Dhakad and Kar [20] have introduced an efficient privacy-preserving data possession with provable security in cloud storage (EPPDP). The EPPDP scheme verifies the data privacy of the owners and supports them by batch auditing and the data possession scheme combines the data from CSP and modifies the data block to generate a precise response. Moreover, EPPDP can preserve the privacy of multiple data owners with minimum overhead but EPPDP is not cost-effective and requires more space to store the data.

Rao [21] have introduced security-aware data transfer in a cloud environment using blowfish algorithm (BFA). Initially, the pattern-matching technique was utilized to identify and import the user data. Moreover, the performance of the BFA was improved using the spotted hyena optimization

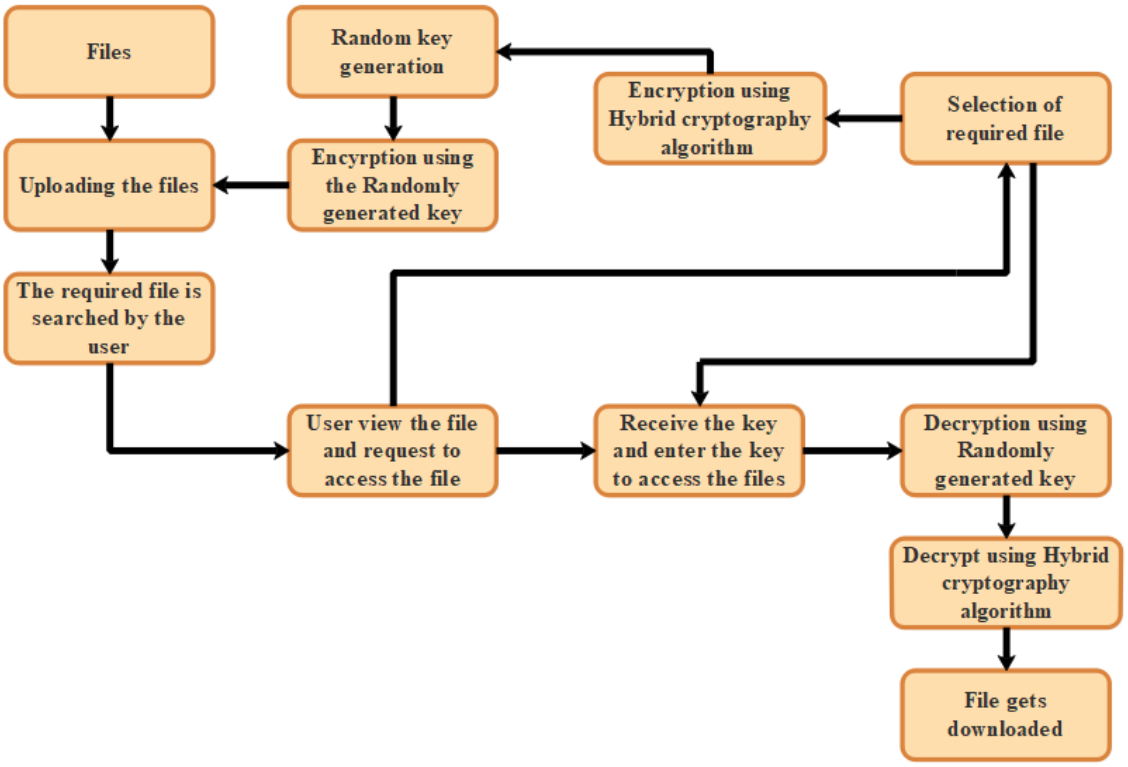

Figure. 1 The process takes place in secured encryption and decryption of files using a hybrid cryptography algorithm

(SHO) algorithm. The improvisation of BFA with SHO helps in the process of authenticating and authorizing the unauthorized client. The suggested BFA-SHO is secured because it only allows the client to access the data after identity detection. However, the execution time of the suggested approach is reliably high.

Jun Ma [22] have introduced a secure data deletion and verification (SDVC) scheme on the basis of ciphertext policy-attribute based encryption (CP-ABE) to secure the deletion and detection verification in the cloud environment. Based on the policy of CP-ABE, the attribute construction tree is utilized as the re-encrypting key to attain the fine grained secured key deletion. The secure cipher text deletion is performed using transposition algorithm which helps in secured key verification.

3. Hybrid cryptography algorithm

Generally, the cryptography algorithms offer a sufficient solution for the issues regarding the confidentiality and secureness of the data. However, for a complex data volume, these algorithms struggle with attacks related to data security. So, this research introduced a hybrid cryptography algorithm by integrating advanced encryption standard (AES), triple data encryption standard (TDES) and blow fish algorithm (BFA). The overall process in the encryption and decryption of data is diagrammatically represented in Fig. 1 as follows:

3.1 Acquisition of data files from the user

Initially, the data is collected from the data owner which includes various information such as personal data, data about the organization, or multimedia-based datasets for analysis of the data. The data of the owner gets stored in the cloud server by using a randomly generated key from the trust center. After this, the information of the users such as the ID of the user, password, and authentication ID is provided. The files of the user, data related to business, images, and video are uploaded to the cloud servers through the internet connection. These aforementioned sensitive data are obtained from the user and stored in the cloud storage space.

3.2 Uploading the data files of the user

The cloud services can be accessed by accomplishing the process of registration such as name, password, and other essentials. The data server of the cloud storage system provides user-specific keys which are utilized in the process of encryption and decryption of the data. When the user starts to upload the file to the cloud server, it gets stored in the temporary folder of the cloud space, and the file of the user gets separated into n parts. These n parts of the user file are encrypted using the proposed hybrid cryptography algorithm which is comprised of

AES,TDES, blowfish, IDEA, and RC6. These algorithms are involved in various processes of data encryption and decryption. After the stage of split encryption, the file is reassembled and stored in the specified folder in the cloud space.

3.3 Key generation

After the stage of uploading the files of the user to the cloud server, random keys are generated to encrypt the user files from security threats. Key generation is defined as the process of creating keys for cryptography where these keys are utilized in the process of encrypting and decrypting the data files of the user. The cryptographic algorithms are categorized into two classes such as symmetric key algorithm and public key algorithm. Generally, the algorithm based on a symmetric key utilizes a single key whereas the public key algorithm uses a public key and private key. Moreover, in the stage of key generation, the generated keys are utilized in the process of encryption and decryption of data files.

3.4 Encryption and decryption of data files using hybrid cryptography

After the stage of key generation, the least significant bit (LSB) method is utilized to embed the secret data. The information of the generated key is encrypted using the proposed hybrid cryptography algorithm. The hybrid cryptography algorithm is comprised of AES, TDES and Blowfish algorithm. This section provides a brief description of the AES, TDES, and blowfish algorithms and the hybridization of those algorithms to provide security in the encryption and decryption of data files in a secure manner.

3.4.1. AES

The AES is a kind of cryptographic algorithm which is utilized in the process of encryption of data which has different key sizes such as 128, 192 and 256 bits. AES is comprised of a continuous series of operations which include input of certain output values and the mixed bits. Every calculation involved in the AES algorithm is presented in terms of bytes instead of bits. So, in AES, 128 bits of data are considered as 16 bytes which is arranged in the matrix form of 4×4. To enhance the security in the encryption of data, the following transformations are performed.

(i) **Substitution bytes:** The data blocks in AES are comprised of 128 bits (i.e. each block is comprised of 16 bytes). In sub byte, every individual 8 bits in a block of the data gets transformed to another data using an 8-bit sub box known as Rijndael S-box.

(ii) **Shifting rows:** The 4 rows of the matrix are rotated in a left-side manner where the outcome obtained from the matrix consists of 16 bytes.

(iii) **Mix columns:** Here, every individual column of the matrix is transformed utilizing the matrix multiplication process. After this, the outcome will be comprised of a new matrix comprised of 16 new bytes. The individual column of the state matrix is considered as the four term polynomial which is multiplied with a specified polynomial represented in Eq. 1 as the matric multiplication.

$$p'(y) = a(y) \times p(y) \tag{1}$$

The matric form of Eq. 1 is represented in Eq. 2 as follows:

$$\begin{bmatrix} P'_{0,c} \\ P'_{1,c} \\ P'_{2,c} \\ P'_{3,c} \end{bmatrix} = \begin{bmatrix} 02\ 03\ 01\ 01 \\ 01\ 02\ 03\ 01 \\ 01\ 01\ 02\ 03 \\ 03\ 01\ 01\ 02 \end{bmatrix} \begin{bmatrix} P_{0,c} \\ P_{1,c} \\ P_{2,c} \\ P_{3,c} \end{bmatrix} \tag{2}$$

(iv) **Add round key:** In this stage, the XOR process is operated in the state of the round key matrix. The add round key transformation is the final transformation of individual round. In the transformation, the round key is obtained based on the state of bitwise operator. The key schedule of every individual round key along with the column is represented as per the Eq. (3) as follows:

$$\left[P'_{0,c}, P'_{1,c}, P'_{2,c}, P'_{3,c}\right] = \left[P_{0,c}, P_{1,c}, P_{2,c}, P_{3,c}\right] \oplus \left[W_{round \times Nb} + c\right] \tag{3}$$

Where the words obtained from the key schedule is represented as $[W_{round}]$ and the round value lies among the range $0 \leq round \leq N_r$ and the round number is represented as N_r.

These aforementioned stages are continued in each round with numbers 10, 12, or 14 based on the key size such as 128, 192, and 256 bits. Moreover, AES algorithm is considered one of the most secure encryption algorithms which secures the significant data files of the user. The AES is known as the fastest encryption algorithm due to its scalability and flexibility with minimized memory usage.

3.4.2. Triple data encryption standard (TDES)

TDES is a kind of cipher-based symmetric

algorithm that is comprised of each block of 64-bit plaintext and employs DES cipher three times to enhance the security of the DES algorithm. Individual 64-bit keys are utilized in every application related to DES. The TDES enhances the security to perform encryption three times slower than DES. The methodology of TDES is defined in Eq. (4) as follows:

$$E^1 = E^3 = E, E^2 = D \tag{4}$$

Where the single encryption function of TDES is denoted as E, the double encryption function of TDES is denoted as E^3 and the decryption is denoted as D. TDES is well suited for both the process of encryption and decryption due to the large sized key lengths. TDES encryption is employed in three ways which are described as follows:

- DES-3EES: Here, 3 types of data encryption standard are used to perform three various types of keys.
- DES-EDES: The three various keys were utilized for three individual processes such as key generation, encryption, and decryption.
- DES-EEE2 and DES-EDES2: The various keys are utilized for the operation of secondary decryption.

3.4.3. Blow fish algorithm (BFA)

The BFA is a kind of symmetric block cipher algorithm that uses a similar key for both the encryption and decryption of data files. BFA is Feistel structured algorithm that consists of cipher blocks and utilizes 64-bit data with 16 round and the key's length fluctuate from 32 bits to 448 bits. Generally, BFA is comprised of two stages as data encryption phase and the key expansion phase. In the latter phase, the length of the variable key is converted to 56 bytes (i.e. 448 bits) array of sub keys and four S-boxes with 32 bits. In the second stage of BFA, the data encryption takes place through 16 Feistel network along with swap and executive operations. In BFA, the F function is utilized to divide the 32-bit input into four equal halves (i.e. 8 8-bit). These values were utilized for table lookup in their respective S boxes.

3.4.4. Encryption using a hybrid of AES, TDES and BFA

In the stage of encryption, the data files are downloaded for the process of encryption then the data files are divided into three slices using file system module. Each slice is encrypted into three phases using AES, TDES, and BFA. After this stage, the slices were combined into an individual file and loaded in the cloud environment. The encryption in the proposed hybrid approach is performed with the help of RSA for the generation of key is listed as follows:

- The RSA algorithm chooses two prime numbers such as p and q then multiplies those numbers to obtain modulus of encryption known as $n = pq$.
- Then choose third number e, which is prime to the product of $(p-1)(q-1)$ and evaluate the integer from the quotient value represented in Eq. (5).

$$I = \frac{(ed-1)}{(p-1)(q-1)} \tag{5}$$

- The public key is the pair of numbers (n, e) which are known publicly It is infeasible to determine d from n and e if p and q are large enough.
- The message is encrypted using public key and cipher text is created is formulated based on the following Eq. (6)

$$C = M^e Mod\ n \tag{6}$$

- After this receiver decrypts cipher text using private key generated with the help of Eq. (7) as follows:

$$M = C^d Mod\ n \tag{7}$$

Where the public exponent is represented as e, the encrypted message is represented as M, and the cipher text that is created with the help of RSA is known as C. Fig. 2 mentioned below presents the diagrammatical representation of the stages involved in the encryption process

3.4.5. Decryption using a hybrid of AES, TDES and BFA

In the process of decryption, the steps are opposite of the process involved in the encryption process. Initially, the encrypted files are divided into three slices as the same in the encryption process and the data files are decoded using the generated key with the proposed hybrid cryptography algorithm. After the decryption performed using the hybrid approach, the generated key is decrypted based on the following steps:

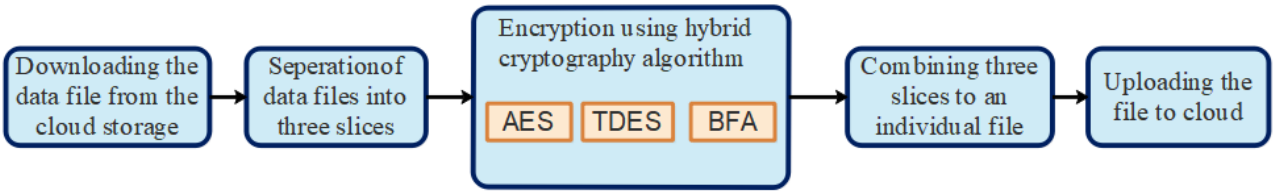

Figure. 2 Encryption using a hybrid cryptography algorithm

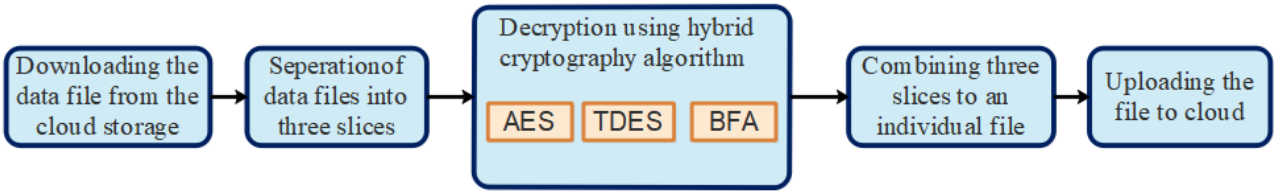

Figure. 3 Decryption using a hybrid cryptography algorithm

- The private key (n, d) obtained from the stage of encryption is evaluated based on the Eq. (8) as follows:

$$M = C^e Mod\ n \tag{8}$$

- After this, the plain text is extracted from the cipher text using the proposed hybrid algorithm for decryption.

Where n is the total number of messages and the encrypted cipher text is represented as C^e. The process involved in the process of decryption is diagrammatically presented in Fig. 3 as follows:

The combination of AES, TDES and BFA helps in effective encryption and decryption. Each slice is encrypted into three phases using AES, TDES, and BFA. After this stage, the slices were combined into an individual file for decryption. In the stage of decryption, the better results of individual algorithms are obtained and fed into RSA for an effective encryption and decryption. The combination of three techniques (i.e. AES, TDES and BFA) provides enhanced security and flexibility during encryption and decryption. Even any one of the algorithms is compromised, the other layers of remaining algorithm provide an additional barrier. The combination of three algorithms tailor the process of encryption and decryption. For example, the speed and strong security is achieved using AES, the compatibility is achieved using 3DES and the blowfish algorithm is known for its efficiency. Though the combination of algorithm results in better security, the key management is a complex process so this research utilized RSA for an effective key management.

4. Results and analysis

This section describes about the obtained results while evaluating the proposed hybrid cryptography algorithm. The suggested algorithm is applied in system which is specified with 8GB random access memory (RAM), intel i7 processor, and Windows 10 OS. Furthermore, the suggested approach is implemented in Python. The effectiveness of the suggested approach is computed using time taken for encryption, decryption, and execution time. The results section is segregated into two sub-sections like performance analysis and comparative analysis.

4.1 Performance analysis

Here, the effectiveness of hybrid algorithm is computed using the time taken to encrypt, decrypt, and execute the text file of various sizes such as 5 MB, 10MB, 15MB, and 20 MB. Moreover, the effectiveness of the hybrid algorithm is related with AES, DES, and blowfish algorithms.

At first, the performance is evaluated by means of encryption time for encrypting the text files with various sizes. Table 1 exhibits the time taken by different algorithms including proposed algorithm to encrypt text files.

The obtained results from Table 1 shows that suggested algorithm took minimum time to encrypt the text files when it is evaluated with the existing algorithms. For an example, the time taken to encrypt 20MB text file is 3100 ms whereas the existing algorithms such as AES, DES and blowfish had taken 3800 ms, 3500, and 3400 ms correspondingly. The

Table 1. Time taken for encryption of text files

Text File Size(MB)	Time taken by AES algorithm (ms)	Time taken by DES algorithm (ms)	Time taken by blowfish algorithm (ms)	Time is taken by the proposed algorithm (ms)
5	3300	3100	3000	2500
10	3500	3200	3100	2800
15	3600	3400	3300	3000
20	3800	3500	3400	3100

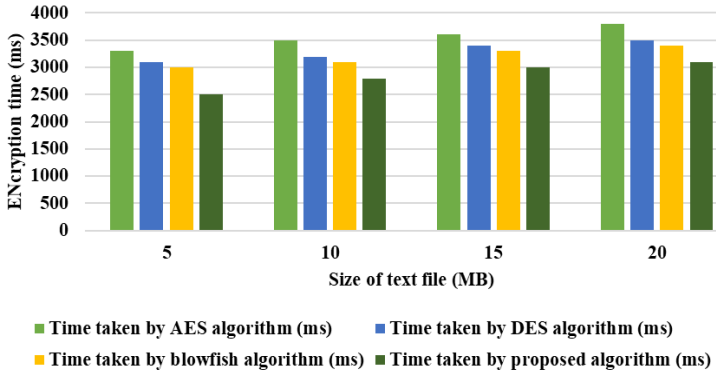

Figure. 4 Graphical representation of time taken to encrypt the text file

Table 2. Time taken for decrypting the data files

Text File Size(MB)	Time taken by AES algorithm (ms)	Time taken by DES algorithm (ms)	Time taken by blowfish algorithm (ms)	Time taken by proposed algorithm (ms)
5	3500	3200	3000	2600
10	3800	3300	3500	2900
15	3900	3500	3400	3100
20	3900	3600	3500	3200

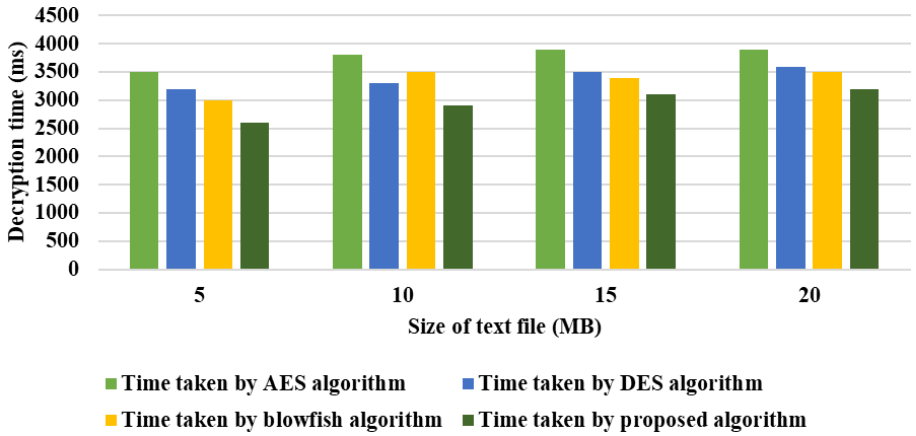

Figure. 5 Graphical representation of time taken to decrypt the text file

obtained results prove the effectiveness of the suggested algorithm in encrypting the text files. The better results are due to the combination of three algorithms which perform the encryption process individually but the existing algorithms encrypt the files in a whole which takes more time and increases the complexity. Fig. 4 presented below shows the graphical representation of the encryption time taken by various algorithms including the hybrid algorithm.

Secondly, the performance is evaluated by means of the time taken to decrypt the text files of different sizes. At the time of decryption, suggested approach comparatively performs better than the existing algorithms and shows better results. The outcome obtained at the time of decrypting the text file is represented in Table 2 as follows:

Table 3. Time taken for execution of the data files

Text File Size(MB)	Time taken by AES algorithm (ms)	Time taken by DES algorithm (ms)	Time taken by blowfish algorithm (ms)	Time is taken by the proposed algorithm (ms)
5	180000	200000	160000	120000
10	210000	210000	180000	140000
15	260000	230000	210000	160000
20	270000	260000	250000	180000

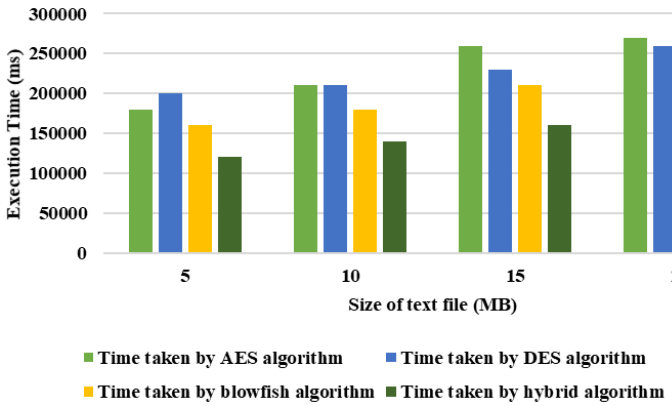

Figure. 6 Graphical representation of time taken to execute the text file

The results from Table 2 exhibits that suggested hybrid algorithm performs well in decrypting the text files with minimum time. The decryption time is evaluated for decrypting the data files using various algorithms such as AES, DES, blowfish, and proposed hybrid of different sizes such as 5MB, 10MB, 15Mb and 20 MB. For example, time taken by hybrid algorithm to decrypt the text file of 20 MB is 3200 ms whereas the existing algorithms such as AES, DES, Blowfish and the proposed hybrid algorithm have taken 3900 ms, 3600 ms, 3500 ms and 3200 ms respectively. The better result is due to the combining the goodness of three algorithms which takes minimum time to decrypt the text files. Fig. 5 presented below shows the graphical representation of the decryption time taken by various algorithms including the hybrid algorithm.

Finally, the performance of the proposed hybrid algorithm is evaluated by means of the time taken to execute the whole process of uploading, encrypting, decrypting and downloading the text files. Table 3 shown below presents the time taken by the algorithms to execute the whole process involved in encryption, decryption, and downloading the data files.

The results from Table 3 show that the execution time taken by the proposed algorithm is relatively lower than the existing techniques. The proposed hybrid algorithm took 180000 ms to execute the text file with the size of 20 MB whereas the existing algorithms such as AES, DES, and blowfish took 270000 ms, 260000 ms and 25000 ms respectively. These obtained results exhibit the efficiency of suggested hybrid algorithm. The better result is due to the minimum time taken by hybrid algorithm in process of encrypting and decrypting the text files. Fig. 6 presented below shows the graphical representation of the execution time taken by various algorithms including the hybrid algorithm.

4.2 Comparative analysis

Here, the results of the suggested approach are evaluated with existing methodologies. The comparison is performed with the existing techniques such as the BFA-SHO algorithm [21] based on the time taken to encrypt, decrypt, and execute the text file. The outcome obtained from the comparison of the existing approach with the proposed hybrid cryptography algorithm is represented in Table 4 as follows:

The result obtained from Table 4 exhibits that suggested hybrid cryptography algorithm has taken minimum time to encrypt, decrypt, and execute text files of various sizes. For example, the execution time of the proposed approach for a file of 20 MB is 183450 ms whereas the existing BFA-SHO takes 248,107 ms respectively.

Secondly, the comparison is performed with the existing TDES [16] based on th execution time, network usage and CPU usage. The results obtained while evaluating the suggested methodology with existing approaches on the basis of fore mentioned metrics is depicted in Table 5 as follows:

Table 4. Comparison of time taken to encrypt, decrypt, and execute the text file of various sizes

Methodologies	Size of text file (MB)	Encryption time (ms)	Decryption time (ms)	Execution time (ms)	Memory utilization (Bits)	Runtime (ms)
BFA-SHO [21]	5	29,874	30,117	187,459	10,567,469	187,459
	10	30,459	32,697	201,464	11,456,987	201,464
	15	32,658	34,793	224,793	12,430,698	224,793
	20	34,986	36,045	248,107	13,579,914	248,107
Hybrid algorithm	5	25035	26,230	124,300	9,477,229	176,238
	10	28304	29,460	144,567	10,126,289	185,105
	15	30430	31,230	160,459	11,302,233	195,165
	20	31450	32,023	183,450	12,223,289	235,102

Table 5. Comparison of execution time, network usage and CPU usage for file size from 100 MB- 500 MB

Methods	Data size (MB)	Execution time (min)	Network usage (GB)	CPU usage (%)
TDES [16]	100	20	0.21	24
	200	25	0.29	24
	300	30	0.36	28
	400	40	0.40	30
	500	55	0.55	33
Hybrid algorithm	100	12	0.18	20
	200	18	0.22	22
	300	24	0.28	25
	400	32	0.32	28
	500	45	0.48	28

The results obtained from the Table 5 shows that the proposed algorithmic approach have obtained better results in overall metrics. For instance, the network usage of existing approach for 500 MB is 0.55 GB whereas the network usage of the proposed approach for 500 MB is 0.48 GB. Similarly, the execution time of the existing approach for 100 MB file is 20 minutes (min) whereas the proposed approach took minimal execution time of 12 min for same file size of 100 MB. These obtained results exhibits the efficiency of the proposed approach.

The final comparison is performed based on the encryption and decryption time for varying size of files from 50 MB to 500 MB. The Table 6 depicted below presents results obtained while evaluating the proposed approach with the existing SDVC scheme based on CP-ABE [22].

The outcome obtained from Table 6 exhibits that suggested approach had taken minimal time to encrypt and decrypt the data files. For instance, the encryption and decryption time of the existing SDVC scheme based on CP-ABE is 4.15 min and 3.00 min for 500 MB respectively. But, the proposed approach took 3.89 min and 2.88 min to encrypt and decrypt the same file size of 500 MB. This obtained result proves the effectiveness of the suggested approach.

Table 6. Comparison of encryption time and decryption time for data size from 50 MB-500 MB

Methods	Data size (MB)	Encryption time (min)	Decryption time (min)
SDVC scheme based on CP-ABE [22]	50	0.01	0.01
	100	0.09	0.03
	150	1.11	0.09
	200	1.25	1.12
	250	2.32	1.57
	300	2.51	1.91
	350	2.94	2.03
	400	3.10	2.27
	450	3.90	2.85
	500	4.15	3.00
Hybrid algorithm	50	0.008	0.01
	100	0.04	0.02
	150	0.80	0.05
	200	1.00	0.8
	250	1.80	1.28
	300	2.10	1.89
	350	2.56	2.20
	400	2.88	2.46
	450	3.25	2.76
	500	3.89	2.88

The better result is due to the usage of algorithms such as AES, TDES and blowfish which individually perform slicing and secure the files in the cloud environment. Moreover, the combination of three techniques (i.e. AES, TDES and BFA) is provided into RSA algorithm where the advantages of those three techniques are obtained and utilized in the process of encryption and decryption.

5. Conclusion

This research introduced an effective hybrid cryptography algorithm to secure the data stored in the cloud environment. The hybridization is performed with advanced encryption standard (AES),

triple data encryption standard (TDES) and blow fish algorithm (BFA). The proposed approach can reliably meet the security requirements of the cloud-based storage system. BFA is utilized in the process of encrypting the slices in minimal time with maximum throughput. The AES and TDES make the cloud server more secure and allow the user to fetch their trust values. The proposed approach utilizes a key to translate the data into unreadable form which allows only the authenticated person to access the files. The experimental results show that the proposed hybrid cryptography technique consumes minimum time to execute the text files, it took 183450 ms to execute 20MB files. However, the existing BFA-SHO took 248,107 ms to execute the same text of 20MB size. In the future, the proposed approach can be implemented in securing the data related to real-time 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] F. Thabit, S. Alhomdy, A. H. A. A. Ahdal, and S. Jagtap, "A new lightweight cryptographic algorithm for enhancing data security in cloud computing", *Global Transitions Proceedings*, Vol. 2, No. 1, pp. 91-99, 2021.

[2] G. Viswanath and P. V. Krishna, "Hybrid encryption framework for securing big data storage in multi-cloud environment", *Evolutionary Intelligence*, Vol. 14, No. 2, pp. 691-698, 2021.

[3] U. Narayanan, V. Paul, and S. Joseph, "A novel system architecture for secure authentication and data sharing in cloud enabled Big Data Environment", *Journal of King Saud University-Computer and Information Sciences*, Vol. 34, No. 6B, pp. 3121-3135, 2022.

[4] Y. M. Gajmal and R. Udayakumar, "Privacy and Utility-Assisted Data Protection Strategy for Secure Data Sharing and Retrieval in Cloud System", *Information Security Journal: A Global Perspective*, Vol. 31, No. 4, pp. 451-465, 2022.

[5] R. Denis and P. Madhubala, "Hybrid data encryption model integrating multi-objective adaptive genetic algorithm for secure medical data communication over cloud-based healthcare systems", *Multimedia Tools and Applications*, Vol. 80, No. 14, pp. 21165-21202, 2021.

[6] K. K. Singh and V. K. Jha, "Security enhancement of the cloud paradigm using a novel optimized crypto mechanism", *Multimedia Tools and Applications*, Vol. 82, No. 11, pp. 15983-16007, 2023.

[7] M. Kamal, S. Amin, F. Ferooz, M. J. Awan, M. A. Mohammed, O. A. Boridi, and K. H. Abdulkareem, "Privacy-aware genetic algorithm based data security framework for distributed cloud storage", *Microprocessors and Microsystems*, Vol. 94, p. 104673, 2022.

[8] K. Sundar, S. Sasikumar, and C. Jayakumar, "Enhanced cloud security model using QKDP (ECSM-QKDP) for advanced data security over cloud", *Quantum Information Processing*, Vol. 21, No. 3, p. 115, 2022.

[9] L. Raji and S.T. Ramya, "Secure forensic data transmission system in cloud database using fuzzy based butterfly optimization and modified ECC", *Transactions on Emerging Telecommunications Technologies*, Vol. 33, No. 9, p. e4558, 2022.

[10] Y. Alemami, A. M. A. Ghonmein, K. G. A. Moghrabi, and M. A. Mohamed, "Cloud data security and various cryptographic algorithms", *International Journal of Electrical and Computer Engineering*, Vol. 13, No. 2, p. 1867, 2023.

[11] S. Guan, C. Zhang, Y. Wang, and W. Liu, "Hadoop-based secure storage solution for big data in cloud computing environment", *Digital Communications and Networks*, In Press, Journal Pre-proof, 2023.

[12] A. E. L. Azzaoui, P. K. Sharma, and J. H. Park, "Blockchain-based delegated Quantum Cloud architecture for medical big data security", *Journal of Network and Computer Applications*, Vol. 198, p. 103304, 2022.

[13] K. K. Singamaneni, A. Juneja, M. A. Elnaby, K. Gulati, K. Kotecha, and A. P. S. Kumar, "An Enhanced Dynamic Nonlinear Polynomial Integrity-Based QHCP-ABE Framework for Big Data Privacy and Security", *Security and Communication Networks*, Vol. 2022, p. 4206000, 2022.

[14] S. Achar, "Cloud Computing Security for Multi-Cloud Service Providers: Controls and

Techniques in our Modern Threat Landscape", *International Journal of Computer and Systems Engineering*, Vol. 16, No. 9, pp. 379-384, 2022.
[15] M. Tahir, M. Sardaraz, Z. Mehmood, and S. Muhammad, "CryptoGA: a cryptosystem based on genetic algorithm for cloud data security", *Cluster Computing*, Vol. 24, No. 2, pp. 739-752, 2021.
[16] M. N. Ramachandra, M. S. Rao, W. C. Lai, B. D. Parameshachari, J. A. 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.
[17] V. Devmane, B. K. Lande, J. Joglekar, and D. Hiran, "Preserving data security in cloud environment using an adaptive homomorphic blockchain technique", *Arabian Journal for Science and Engineering*, Vol. 47, No. 8, pp. 10381-10394, 2022.
[18] K. Sundar, S. Sasikumar, and C. Jayakumar, "Enhanced cloud security model using QKDP (ECSM-QKDP) for advanced data security over cloud", *Quantum Information Processing*, Vol. 21, No. 3, p. 115, 2022.
[19] P. Shrivastava, B. Alam, and M. Alam, "Security enhancement using blockchain based modified infinite chaotic elliptic cryptography in cloud", *Cluster Computing*, 2022.
[20] N. Dhakad and J. Kar, "EPPDP: An Efficient Privacy-Preserving Data Possession With Provable Security in Cloud Storage", *IEEE Systems Journal*, Vol. 16, No. 4, pp. 6658-6668, 2022.
[21] C. C. Rao, T. Hiwarkar, and B. S. Kumar, "Cloud-based data security transactions employing blowfish and spotted hyena optimisation algorithm", *Journal of Control and Decision*, Vol. 10, No. 4, pp. 494-503, 2022.
[22] J. Ma, M. Wang, J. Xiong, and Y. Hu, "Cp-abe-based secure and verifiable data deletion in cloud", *Security and Communication Networks*, Vol. 2021, pp. 1-14, 2021.