JCSDA, Vol. 1, No. 2, 34–58
DOI: 10.69660/jcsda.01022403
ISSN 2959-6912

Improving Data Security in Cloud Storage Systems Using Hybrid Algorithms with Integrity Verification

Erku Kifle Desie

Dep't of Software Engineering,
Addis Ababa Science and Technology University,
Addis Ababa, Ethiopia

Asrat Mulatu Beyene

Big Data Analytics & High Performance Computing CoE,
Dep't of Electrical and Computer Engineering,
Addis Ababa Science and Technology University, Addis Ababa, Ethiopia
Corresponding author: asrat.mulatu@aastu.edu.et

In a cloud system, storage services allow users to store data on the Internet. However, storing data in the public cloud increases the risk of loss, interception, modification, and manipulation by unauthorized users. So, users need to protect their data by applying security mechanisms. In this work, we proposed an efficient hybrid algorithm along with data slicing and integrity verification to improve user-side cloud data security. The proposed hybrid algorithm combines multiple symmetric and asymmetric algorithms to improve both performance and security. It takes the advantages of both to compensate for their weaknesses. The asymmetric algorithm is used for encrypting the symmetric keys, whereas, the symmetric algorithms are used to encrypt and decrypt the data. The proposed framework works by splitting the data into chunks and encrypting each portion separately. File information such as type of algorithm used, hash value, and secret keys are kept at the user's side enabling only the encrypted data to be sent to the cloud. The proposed hybrid algorithm is evaluated and compared against the-state-of-the-art. The results show that the proposed hybrid algorithm outperformed existing ones in terms of throughput, and running time while achieving better degree of data security.

Keywords*: Hybrid Cryptographic Algorithm, Cloud Data Security, Security Performance Evaluation, Asymmetric Algorithm, Symmetric Algorithm, Data Integrity Verification.*

1. Introduction

Cloud computing is network-based computing that provides shared resources to its users whenever required [1]. In cloud computing, data security is one of the most critical concerns due to sensitivity of data stored in the cloud. To protect data from different hackers, cryptography is one of the popular methods. The confidentiality, integrity, and availability of sensitive data in the cloud become a very critical problem since it is not maintained in user devices [2]. Some unauthorized users or cloud storage service providers may manipulate or modify data without the knowledge of users since cloud users do not have control of the data once it is sent to the cloud storage server [2]. Cloud storage allows cloud users to save data and files in an off-site location so that it is accessed latter through the Internet.

The proposed framework improves the security of data using hybrid cryptographic algorithms along with integrity verification. The hybrid algorithms are constructed from existing symmetric and asymmetric security algorithms. The performance of encryption and decryption processes of large data has been improved. The time taken for encryption and decryption, and the footprint needed to store large data are significantly improved. Moreover, when the encrypted data are downloaded and decrypted, the user can check the integrity of the data, followed by merging the partitioned data to get the original file.

The hybrid algorithms constructed from symmetric and asymmetric algorithms compensate for their individual weaknesses. In the symmetric algorithm, constructing a hybrid algorithm from more than one algorithm strengthens its security making it more unbreakable by an unauthorized user. However, in symmetric algorithm, the secret key is the same for encoding and decoding the data. This leads to the sharing of the same secret key among multiple users which faces security risks. An asymmetric encryption algorithm uses two different keys to encode and decode data. This achieves better security of data since it doesn't share keys. Of course, it introduces more delay in encrypting and decrypting larger data. To solve these challenges the proposed hybrid algorithm takes the advantages of both symmetric and asymmetric algorithms to improve the security of storing data in a cloud storage platform.

2. Related Works

There are many works that have tried to improve the security of network communications system, data storage, and different kinds of files in cloud systems. Sonia Rani and Harpreet Kuar in [3], proposed a hybrid encryption model to secure network communication using AES and ElGamal algorithm. The implementation of this model shows that the time required for encrypting and decrypting using a hybrid of AES and ElGamal is less than the individual AES and ElGamal algorithms. The hybrid algorithm is better than the individual ones in terms of encryption and decryption times.

S.V.N. Srivalli and Ben Swarup Medikonda in [4] developed cloud-based secure text file application using a hybrid of cryptography and steganography techniques. The peculiarity of their work is that the text file is split into two equal parts and the first part is encrypted with AES and the second with blowfish algorithm. The secret keys of both algorithms are hidden with the help of steganography with Least Significant Bit (LSB) algorithm.

Bhushan Rathod and Prashant Yelmar [5], proposed an efficient data security mechanism to prevent insider threats of cloud storage systems based on the AES in combination with different hashing algorithms. It has provided a preventive and proactive measure to defend against different internal security threats.

Ali Abdulridha Taha et al. [6], developed a new hybrid cryptography algorithm for data security in the cloud computing storage system by constructing from ex-

isting algorithms. The algorithms they used to develop new hybrid algorithms are blowfish, AES, Krishina, RSA, and triple DES. The proposed hybrid algorithms provide good security to data and minimize the time for encryption and decryption processes. The performance of developed hybrid algorithms was compared with each other based on encryption time, decryption time, and throughput.

Vimala M. and Sri Preethaa K.R [7], proposed an adaptive multilevel data security protection mechanism for the cloud storage system. In this framework, the organization data is classified as highly sensitive data, moderate sensitive data, and low sensitive data. Here the model assigned a security algorithm for each classified data based on its degree of sensitivity. Then to secure each data, the author applied adaptive multilevel data security techniques based on cryptographic algorithms.

Mihir Shah [8] proposed a hybrid cryptosystem for securing data in the cloud storage system. The designed hybrid cryptography was enhanced with the additional layer of security by combining AES, DES, RC6, ECB, CBC, Triple DES algorithms. The proposed approach helps in reducing encode and decode time and helps in improving the performance of storing large data files in a highly secure environment.

Ponnuru Sowjanya and K. V. N. Sunitha [9], proposed enhanced security of data using a hybrid cryptography algorithm. The hybrid algorithm they used was a combination of symmetric blowfish algorithm and asymmetric RSA algorithm. The hybrid algorithm makes the cipher data stronger and difficult to convert into its original form.

Shilpi Harnal and R.K. Chauhan [10], proposed a hybrid cryptography-based end-to-end encryption for keeping the integrity and confidentiality data. The author implements a hybrid algorithm that combines one symmetric algorithm and one asymmetric algorithm. The modified blowfish algorithm was used to encrypt the data and the public key algorithm was used to maintain the confidentiality and authentication of the sender and receiver.

Diaa Salama AbdElminaam [11] **,** proposed improved security of cloud computing by building new hybrid cryptography algorithms from the existing AES, RSA, Blowsh, and MD5 cryptographic algorithms. Here, the author concludes that the hybrid algorithm increases the security of the key plus increasing security in which the cipher-text cannot be decrypted except by the recipient. The parameters the author used were encryption time, decryption time, size of cipher-text, and throughput of the hybrid algorithm. Chandrika and Sahil Dalwal [12], proposed a data security technique using hybrid algorithms by combining DES, AES, and RSA algorithms. The author proposed a three-layer encryption mechanism and first, the plain text was encrypted by using the DES algorithm, and the ciphered data was again encrypted using the AES algorithm and finally, this ciphered data was encrypted using RSA algorithm to increase the degree of security to protect from unauthorized use. And the decryption process was performed as the reverse process of encryption.

Dhuratë Hyseni and Artan Luma [13], proposed a model to increase the security

of sensitive data in cloud computing based on hybrid algorithms of AES, DES, and ElGamal. The framework provides an environment to select data sensitivity levels from highly sensitive data to low sensitive data. After selecting levels of data sensitivity, the user would select the algorithms used to encrypt the data and then send it to the cloud or download it from the cloud.

Aditya Poduval et al. [14], proposed a secure file storage system for cloud computing platforms using hybrid cryptography algorithms. The algorithms they used were 3DES, AES, RC6, and steganography, and securely store and retrieve data was under the control of the owner. Key information was securely stored using the LSB technique and data was secured using a combination of AES, RC6, and 3DES algorithms. The proposed framework provides better data integrity, high security, low delay, authentication, and confidentiality.

K. Subramanian and F. Leo John [15], proposed enhanced security for storing and sharing data in a multi-cloud storage system using AES algorithm. The proposed system ensures the protection of data from malicious users and reduces the risk of an insider attack on the stored data. It supports all file formats and the slice and merges mechanism is also applied to the uploaded file to store in multiple data storage servers.

P. Chinnasamy and P. Deepalakshmi [16], designed a secure cloud storage system for health-care using a hybrid cryptography algorithm. The algorithms used were blowfish for encrypting the data and RSA for encrypting the secret key of blowfish to send and receive data securely.

Neha and Mandeep Kaur [17], proposed enhanced data security using a hybrid encryption algorithm. The hybrid encryption algorithm was constructed from the existing symmetric algorithms of AES, Blowfish, and Twofish. The proposed system considers encryption and decryption time as a metric to compare the hybrid AES and Twofish algorithm with that of AES and blowfish algorithm. The experiment shows that AES with Twofish takes less time than AES with the Blowfish algorithm.

Moses Okechukwu Onyesolu and Nwachukwu Chigozie Nnabugwu [18], introduced an enhanced data security approach based on hybrid AES and RSA encryption algorithm with SHA512. The use of a hybrid encryption algorithm takes the advantages of both algorithms and provides a more reliable and efficient data security mechanism which is exhibited in this work.

V. Kapoor and Rahul Yadav [19], proposed a hybrid cryptography technique for improving network security. The hybrid algorithm was constructed and developed by hybridizing RSA, SHA12, and DES algorithms. The proposed hybrid approach improves security by generating and checking the integrity of data at both ends of the network. It is effective and essential for network data security where the two users are communicating in unknown networks.

Munavvara Tahaseen et al.[20] **,** designed a data storage framework for cloud platforms using hybrid encryption with a one-time password technique. It uses a combination of symmetric and asymmetric algorithms to take the advantage of security and speed. The symmetric key was used to encrypt the message and the

public key was used to encrypt the secret symmetric key and the asymmetric private key was used to decrypt the encrypted key. They also add OTP (one-time-password) feature for outsourcing data to the cloud storage service provider. In this work, the security of the key was safe, and user authentication was simple and secure.

Joseph Selvanayagam et al. [21], proposed a secure file storage system for storing data in the cloud using a cryptography algorithm. They use elliptic curve cryptography encryption to protect data files in the cloud. The system ensures the security and privacy of client sensitive information by storing data across a single cloud using AES, DES, and RC2 algorithm. Rohit Barvekar [22], proposed hybrid cryptography on cloud environment using AES and RSA algorithm. The author compares the result of the original RSA and hybrid AES & RSA algorithms in terms of running time. Since the running time of an improved hybrid RSA and AES was increasing than the original RSA algorithm. This hybrid algorithm is helped to prevent brute force, mathematical, and timing attacks.

Ahmad Habboush [23] **,** proposed a multi-level data security framework by constructing a hybrid algorithm. The proposed framework combines the strength of Feistel, AES, Crossover & mutation, and HMAC. The framework was evaluated against symmetric encryption algorithm RC6, DES, 3DES for performance, and security metrics. The author concludes that the proposed framework has the lowest running time, higher throughput, and passes the avalanche effect criterion.

In many of these works even though multiple algorithms are used chopping down the original data in to chunks and encrypt each separately was not considered. Moreover, in this work the metadata of the data is kept with the user and only the encrypted data is sent to the cloud storage. Above all, when the user wants to use the data, after downloading its integrity is checked by using the hash value generated before uploading the data to the cloud. This hash value, together with other metadata of the original data, is kept with the user.

The summary on related works is given in Table 1.

3. Proposed Framework

The proposed architecture, depicted in Fig. 1, comprises of the following components and operations:

Uploading data – the user browses any type of file from the local machine that will be uploading to the cloud storage system through the proposed framework.

Splitting data – this is the process of partitioning or slicing the original uploaded file into several chunks based on user needs and the available cloud server.

Message digest ($\mathbf{MD}_e$) – this function calculates the message digest of each sliced parts before encryption.

Encryption of data – it is a process of converting the original readable file into a meaningless file format.

Storing data – after encryption of each data part, the data will send to the cloud and the sender data is stored in the cloud storage platform.

Table 1. Summary of Related Works

Author	**System Considered**	**Item Secured**	**Proposed Algorithm**
Srivalli [4]	Cloud App	Text File	Hybrid of AES, Blowfish & Steganography (LSB)
Bhushan Rathod et al. [5]	Cloud Storage	Data	Hybrid of AES & Hash
Ali Abdulridha Taha et al. [6]	Cloud Storage	Data	Hybrid of Blowfish, AES, Krishna, RSA & 3DES
Vimala M. et al. [7]	Cloud Storage	Data	Adaptive multilevel security algorithm
Mihir Shah [8]	Cloud Storage	Data	Hybrid AES, DES, RC6, ECB, CBC, & 3DES
Ponnuru Sowjanya et al. [9]	Cloud Platform	Data	Hybrid of Blowfish and RSA
Shilpi Harnal et al. [10]	Multimedia Cloud Computing	Data	Hybrid of Blowfish & Symmetric Secure Key
Diaa S. AbdElminaam [11]	Cloud Computing	Data	Hybrid of AES, RSA, Blowfish and MD5
Shakeeba S. Khan et al. [12]	Cloud Computing	Data	Layers of DES and AES
Dhurate Hyseni et al. [13]	Cloud Computing	Data	Layers of AES, DES and ElGamal
Aditya Poduval et al. [14]	Cloud Computing	File	Hybrid of 3DES, AES, RC6, and LSB
K. Subramanian et al. [15]	Multi-Cloud Storage	Data	AES
Chinnasamy et al. [16]	Cloud Storage	Data	Hybrid Blowfish and RSA
Neha et al. [17]	Cloud Computing	Data	Hybrid of AES, Blowfish, and Twofish
Nadesh R.K. et al., [18]	Cloud Storage	Data	Hybrid of AES, and HMAC-SHA
V. Kapoor et al. [19]	Public Networked System	Data	Hybrid of RSA, SHA2, and DES
Munavvara Tahaseen et al. [20]	Cloud Data Storage	Data	File index & encrypted symmetric keys with OTP
J. Selvanayagam et al. [21]	Cloud Storage	Data	Hybrid of AES, DES and RC2
Rohit Barvekar [22]	Cloud Storage	Data	Hybrid of AES and RSA
Ahmed Habboush [23]	-	-	Hybrid of Feistel, AES, HMAC, crossover & mutation
M. N. Abdul Wahid et al. [24]	Cloud Computing	Data	Comparative analysis
Theda Flare G. et al. [25]	-	Data	Modified Blowfish
Hossein Abroshan [26]	Cloud Computing	Data	Hybrid of Blowfish, Elliptic Curve, & digital signature
Ting Yuan Nie et al. [27]	Networked Systems	Data	Performance evaluation of DES and Blowfish
Salim Ali Abbas et al. [28]	Cloud Computing	Data	RC6
Mustafa S. Abbas et al. [29]	Cloud Computing	Data	Hybrid of AES, RSA, LSB and SHA
Fursan Thabit et al. [30]	Cloud Computing	Data	Feistel & substitution permutation architecture
Anjali D.V. et al. [31]	Cloud Computing	Data	Role-based Hybrid AES, RSA, and SHA-1

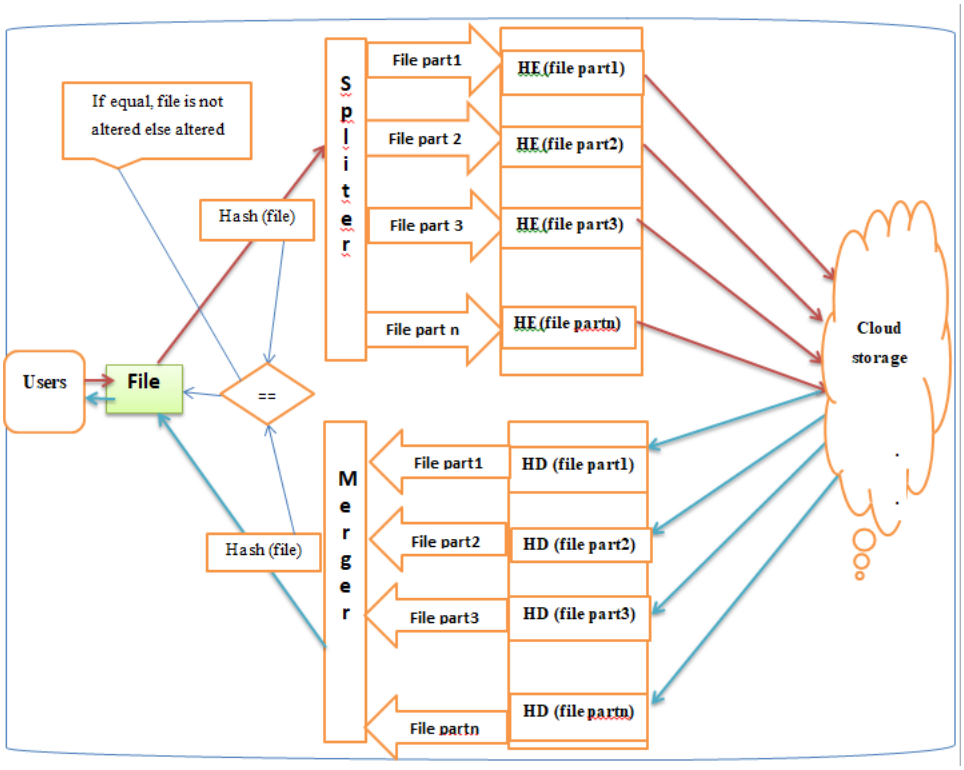

Fig. 1. The Proposed Framework

Downloading data – the data stored in each cloud server will be downloaded based on the user's need. The downloaded file must be equal to the original file by comparing the hash values after decryption.
Decryption of data – after downloading each cipher data, the decryption process for each cipher data will be performed to get the original data parts.
Message digest (MD_d) – when each sliced part is decrypted and joined to form the original file, the sliced original file part must be hashed with a message-digest algorithm. To check the integrity of each sliced parts, the MD_e of each file must be compared with the MD_d of the decrypted sliced file parts. And, if both message digests are equal or the same, the integrity of sliced data is ensured.
Merging data – the merging process takes place to join the sliced file parts to get the original data.

The proposed framework is given in Fig 1

Table 2. Security Algorithms Considered

	Nature	**Architecture**	**Key Size (bits)**	**Strength**	**Public or Patented**	**Known Attacks**
AES [23] [24]	Block cipher, Multimodal, Symmetric	Substitution, Permutation	128, 192, or 256	Very fast, stronger	Public	Inverse cipher implementation on smart cards
Blowfish [25]	Block cipher, Symmetric	Feistel	128 – 448	Faster	Public	2^{nd} order differential attacks
RC6 [27]	Block cipher, Symmetric	Feistel	128, 192, 256	Very fast	Patented	Correlation
RC4 [32]	Stream cipher, Symmetric	Feistel	1 – 256	Very fast	Patented	BEAST
RSA [24]	Block cipher, Asymmetric	Factorization	1024, 2048, or 4096	Slower	Now, public	Factoring the public key, Cycle attack
MD5 Hash Function [33]	Block cipher, message digest	Compression with Merkle-Damgård		Fast and simple	Public	Dictionary attacks, Rainbow tables

4. Methodology

4.1. *Security Algorithms Used*

In this work five different security algorithms are used in four different combinations to investigate the data and key security keeping a critical evaluation of performance. In addition MD5 based hash function is used to keep the integrity of the data. All of these are described briefly in Table 2.

4.2. *Hybrid Algorithms Considered*

4.2.1. *Blowfish – RC4 Hybrid Algorithm with RSA*

This hybrid algorithm, depicted in Fig. 2, combines the RC4 encryption algorithm with that of the Blowfish algorithm to secure data in the cloud storage system. The combined algorithm's key length is longer than the individual algorithm's key. The generated secret key of both algorithms is encrypted using RSA asymmetric algorithm to store secret keys safely.

4.2.2. *Blowfish – AES Hybrid Algorithm with RSA*

This hybrid algorithm, shown in Fig. 3, combines the AES encryption algorithm with that of the Blowfish algorithm to secure data in the cloud storage system. The combined algorithm's key length is longer than the individual algorithm's key.

The generated secret key of both algorithms is encrypted using RSA asymmetric algorithm to store secret keys safely.

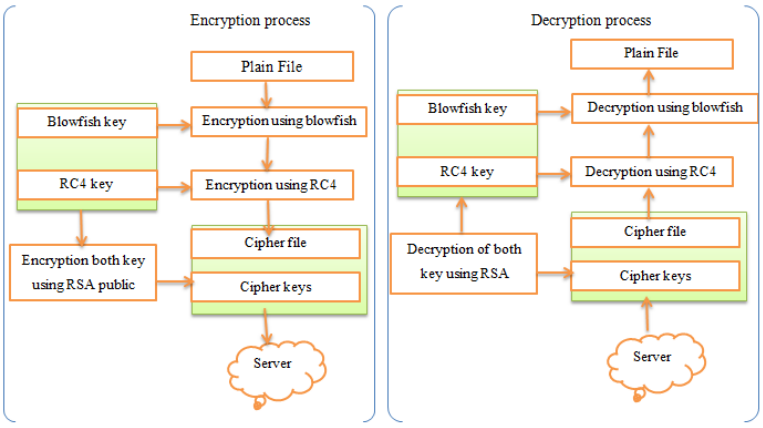

Fig. 2. Hybrid encryption algorithm using Blowfish and RC4 with RSA

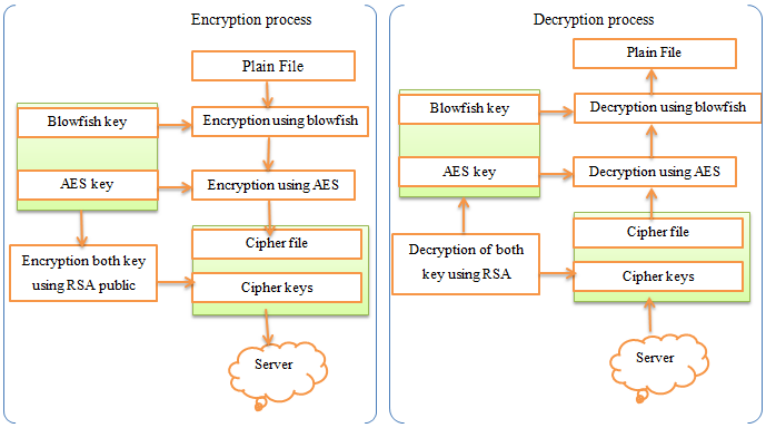

Fig. 3. Hybrid encryption algorithm using Blowfish and AES with RSA

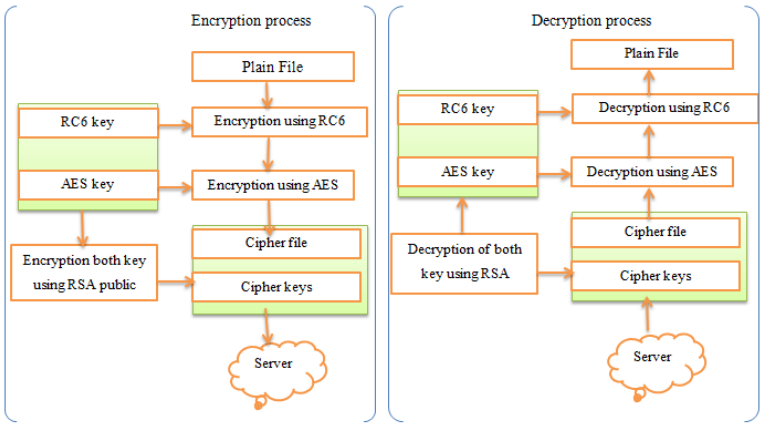

Fig. 4. Hybrid encryption algorithm using AES and RC6 with RSA

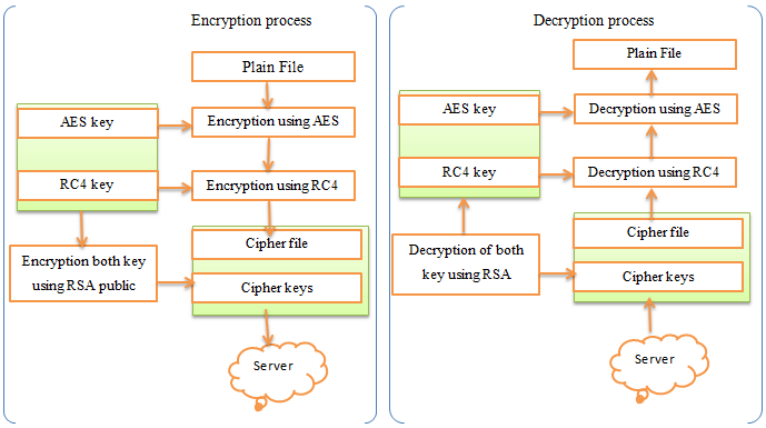

Fig. 5. Hybrid encryption algorithm using AES and RC4 with RSA

4.2.3. *AES – RC6 Hybrid Algorithm with RSA*

This hybrid algorithm depicted in Fig. 4 combines the AES encryption algorithm with that of the RC6 algorithm to secure data in the cloud storage system. The combined algorithm's key length is longer than the individual algorithm's key.

4.2.4. *AES – RC4 Hybrid Algorithm with RSA*

The hybrid algorithm shown in Fig. 5 combines the AES encryption algorithm with that of the RC4 algorithm to secure data in the cloud storage system. The combined algorithm's key length is longer than the individual algorithm's key. The generated secret key of both algorithms is encrypted using RSA asymmetric algorithm to store secret keys safely.

5. Results and Discussion

A prototype of the proposed framework shown in Fig. 1 is implemented using the Azure blob storage cloud server with the C# Windows application. All experiments are conducted in a Windows-based machine that is equipped with 8 GB of memory and an Intel i5 8250U 1.6 GHz CPU. Encryption time, decryption time, and throughput are the performance metrics used.

5.1. *Results*

Four different file formats, namely .csv, .pdf, .pptx and .mp4, ranging in size from 100 MB to 1000 MB are used to test the proposed framework.

(A) File Slicing and Merging

File slicing is a process of partitioning a file into several small and equal-sized parts and File merging is a process of joining or combining sliced parts of a file into the previous original file. Different file formats range from 100 MB to 1000 MB sizes are used to test the proposed work.

Results are given on Tables 3 - 6 and Figs. 6 & 7

Table 3. Slicing and merging time for video files

No.	File type	File size in MB	Slicing time in Second	Merge time in Second
1	.mp4	100	0.339	0.316
2	.mp4	200	5.047	1.117
3	.mp4	300	8.475	1.199
4	.mp4	400	6.222	1.992
5	.mp4	500	10.407	4.108

(B) Encryption and Decryption Time

Table 7, Fig. 8, and Fig. 9 show the time taken to encrypt and decrypt different sizes of .mp4 files. In the same token,

Table 8, Fig. 10 and Fig. 11

Table 9, Fig. 12 and Fig. 13

Table 10, Fig. 14 and Fig. 16

Table 4. Slicing and merging time for PDF files

No.	File type	File size in MB	Slicing time in Second	Merge time in Second
1	.pdf	100	1.002	0.337
2	.pdf	200	3.978	0.610
3	.pdf	300	6.754	0.884
4	.pdf	400	11.902	1.912
5	.pdf	500	13.582	3.090
6	.pdf	1000	37.981	10.639

Table 5. Slicing and merging time for CSV files

No.	File type	File size in MB	Slicing time in Second	Merge time in Second
1	.csv	100	2.990	0.326
2	.csv	200	4.765	0.603
3	.csv	300	7.072	0.962
4	.csv	400	10.108	2.253
5	.csv	500	12.588	3.249

Table 6. Slicing and merging time for PPTX files

No.	File type	File size in MB	Slicing time in Second	Merge time in second
1	.pptx	100	1.605	0.323
2	.pptx	200	4.347	0.614
3	.pptx	300	8.172	0.895
4	.pptx	400	10.359	2.096
5	.pptx	500	11.182	2.753
6	.pptx	1000	28.189	9.607

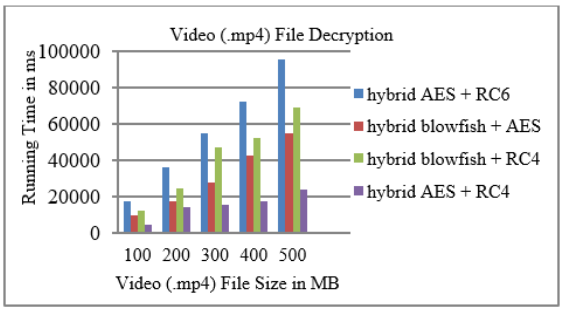

Fig. 9. Decryption Time for Video (.mp4) Files

Table 7. Encryption and decryption time for video (.mp4) files

No.	File Type	File Size (MB)	Encryption Time in Second				Decryption Time in Second			
			Blowfish + RC4	AES + RC4	AES + RC6	Blowfish + AES	Blowfish + RC4	AES + RC4	AES + RC6	Blowfish + AES
1	.mp4	100	11.323	4.467	17.66	8.587	12.226	4.567	17.57	9.926
2	.mp4	200	23.368	8.196	36.42	18.358	24.338	14.433	36.16	17.660
3	.mp4	300	37.548	16.819	53.72	26.477	47.189	15.744	54.45	27.499
4	.mp4	400	48.982	19.582	71.35	42.224	52.006	17.334	72.15	42.511
5	.mp4	500	69.462	25.981	93.00	47.564	69.171	23.886	95.20	54.811

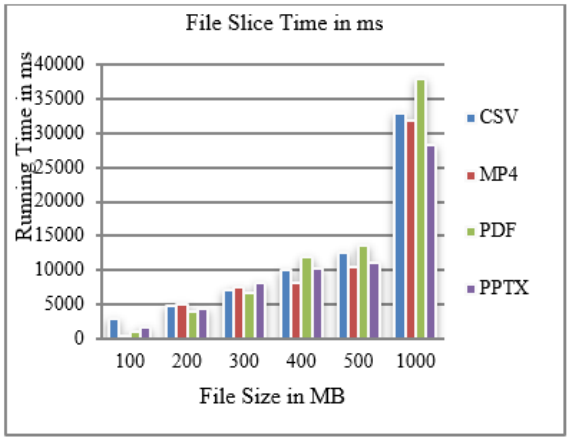

Fig. 6. File slicing time in millisecond of different files

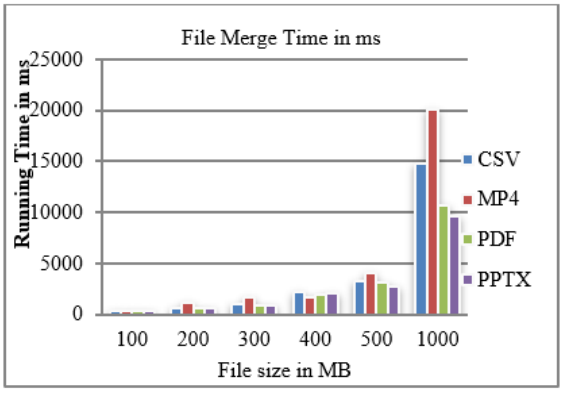

Fig. 7. Files merging time in millisecond of different files

Table 8. Encryption and decryption time for PDF files

No.	File Type	File Size (MB)	Encryption Time in Second				Decryption Time in Second			
			Blowfish + RC4	AES + RC4	AES + RC6	Blowfish + AES	Blowfish + RC4	AES + RC4	AES + RC6	Blowfish + AES
1	.pdf	100	12.353	3.711	17.999	9.377	11.325	7.764	16.83	9.121
2	.pdf	200	24.039	9.920	36.604	20.054	25.218	14.691	35.30	18.259
3	.pdf	300	39.055	14.849	53.514	27.989	37.988	17.112	56.35	30.759
4	.pdf	400	47.212	20.608	70.887	39.521	49.533	22.709	69.69	40.542
5	.pdf	500	74.037	31.028	85.881	49.724	61.959	29.472	93.82	53.712

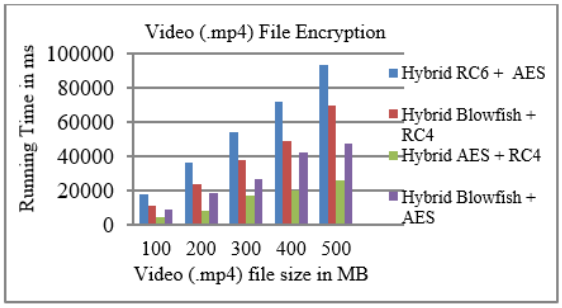

Fig. 8. Encryption Time for Video (.mp4) Files

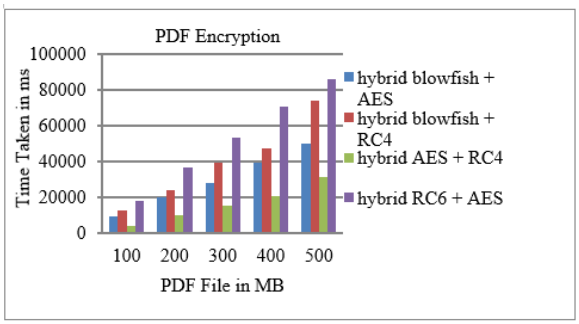

Fig. 10. Encryption time for PDF files

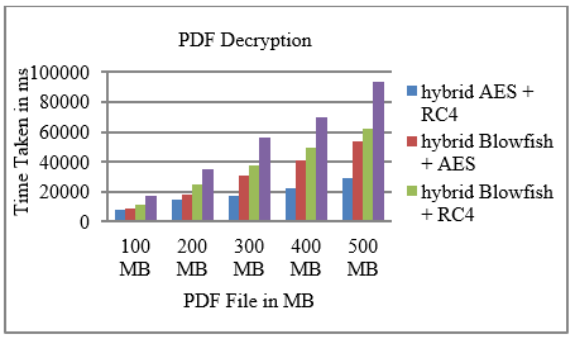

Fig. 11. Decryption time for PDF files

(C) Throughput

Throughput was calculated by dividing the size of the file in bytes over the consumed time in seconds [6] . The throughput of the algorithm determines the speed of the algorithm during encryption and decryption pro-

Table 9. Encryption and decryption time for .csv files

No.	File Type	File Size (MB)	Encryption Time in Second				Decryption Time in Second			
			Blowfish + RC4	AES + RC4	AES + RC6	Blowfish + AES	Blowfish + RC4	AES + RC4	AES + RC6	Blowfish + AES
1	.csv	100	10.970	3.588	18.165	9.197	10.801	3.945	18.372	9.111
2	.csv	200	23.426	8.178	35.934	21.200	24.049	8.179	36.404	21.838
3	.csv	300	35.279	15.393	50.686	33.965	37.144	13.968	53.064	28.677
4	.csv	400	55.958	21.923	72.19	39.494	50.386	19.135	75.783	37.331
5	.csv	500	64.416	26.492	92.553	49.675	62.911	29.606	85.985	59.673

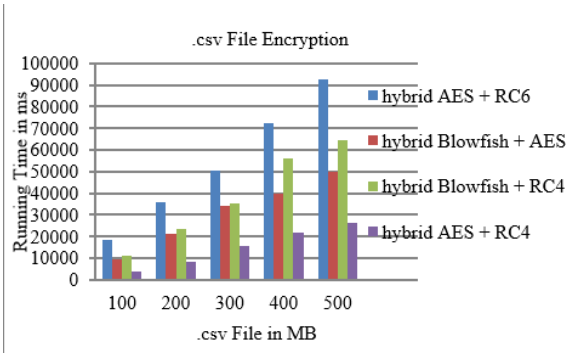

Fig. 12. Encryption time for .csv files

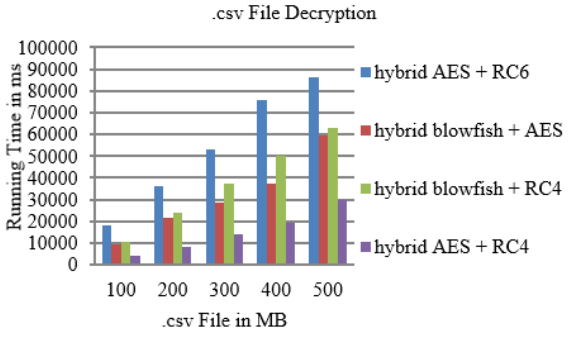

Fig. 13. Decryption time for .csv files

cesses.

5.2. *Comparative Analysis of Results*

From the constructed four hybrid algorithms, the experimental results show that the hybrid AES and RC4 algorithm with RSA is better than hybrid of Blowfish

Table 10. Encryption and decryption time for PPTX files

	File Type	File Size (MB)	Encryption Time in Second				Decryption Time in Second			
			Blowfish + RC4	AES + RC4	AES + RC6	Blowfish + AES	Blowfish + RC4	AES + RC4	AES + RC6	Blowfish + AES
1	.pptx	100	11.455	3.458	18.04	8.843	10.722	3.567	18.29	8.714
2	.pptx	200	25.081	10.071	38.55	18.150	28.648	10.30	35.42	18.51
3	.pptx	300	40.821	14.893	51.13	27.242	41.661	14.71	54.28	27.45
4	.pptx	400	49.752	21.625	68.70	37.349	50.481	18.44	70.04	40.42
5	.pptx	500	67.864	25.198	92.92	47.787	66.419	29.06	95.34	54.21

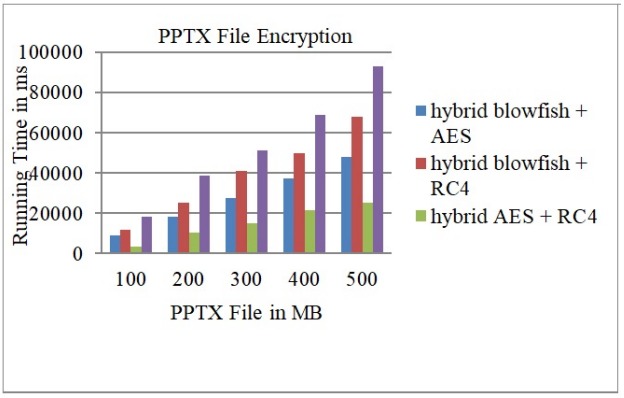

Fig. 14. Encryption time for PPTX files

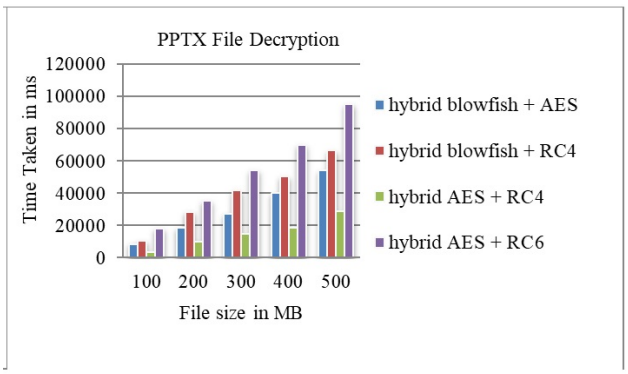

Fig. 15. Decryption time for PPTX files

and AES with RSA, hybrid of RC6 and AES with RSA, and hybrid of RC4 and Blowfish with RSA based on encryption time, decryption time, and throughput. So, the better hybrid algorithm of AES and RC4 with RSA is chosen and this hybrid

Table 11. Throughput of Encryption for PPTX Files

	File Type	**File Size in MB**	**Throughput**			
			Hybrid AES + RC4	**Hybrid Blowfish + AES**	**Hybrid AES + RC6**	**Hybrid Blowfish + RC4**
1	.Pptx	100	29652.69	11595.5	5683.35	8951.462
2	.Pptx	200	20390.73	11314.33	5327.393	8187.672
3	.Pptx	300	20655.14	11292.01233	6017.076129	7535.753656
4	.Pptx	400	18966.8	10981.7	5970.25	8244.01
5	.Pptx	500	20339.95	10725.22	5515.955	7552.252

Table 12. Throughput of Encryption for PDF Files

	File Type	**File Size in MB**	**Throughput**			
			Hybrid AES + RC4	**Hybrid Blowfish + AES**	**Hybrid AES + RC6**	**Hybrid Blowfish + RC4**
1	.pdf	100	27726.49	10972.91	5716.595	8329.394
2	.pdf	200	20744.56	10261.59	5621.954	8560.506
3	.pdf	300	20787.86	11028.58	5768.192	7903.700
4	.pdf	400	19885.29	10369.07	5780.975	8679.912
5	.pdf	500	16557.40	10331.89	5982.033	6939.003

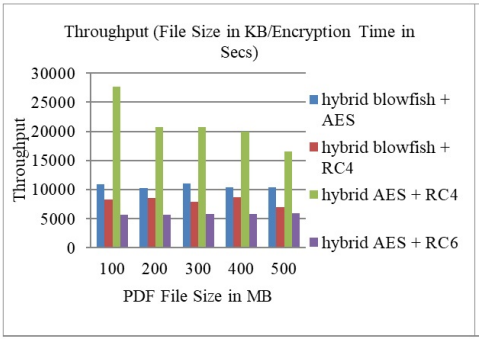

Fig. 16. Throughput of Encryption of PDF Files

algorithm is then compared with previous works.

Scenario 1: File splitting and merging running time comparisons of the proposed framework with existing hybrid security algorithms with file sizes of 52, 212, 345, 437, and 550 MB are as follows.

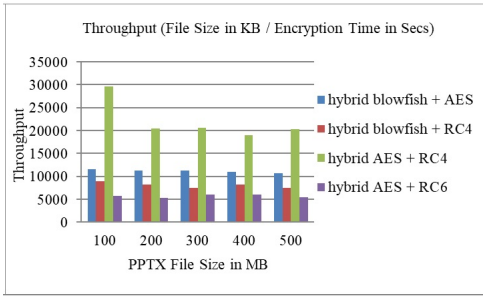

Fig. 17. Throughput for PPTX Files

Table 13. Throughput of Encryption for CSV Files

	File Type	**File Size in MB**	**Throughput**			
			Hybrid AES + RC4	**Hybrid Blowfish + AES**	**Hybrid AES + RC6**	**Hybrid Blowfish + RC4**
1	.csv	100	28578.32	11149.18	5644.867	9347.22
2	.csv	200	25055.76	9665.377	5702.288	8746.948
3	.csv	300	19952.51	9042.514	6059.444	8705.717
4	.csv	400	18682.11	10370.39	5673.473	7319.204
5	.csv	500	19355.35	10322.34	5540.199	7960.165

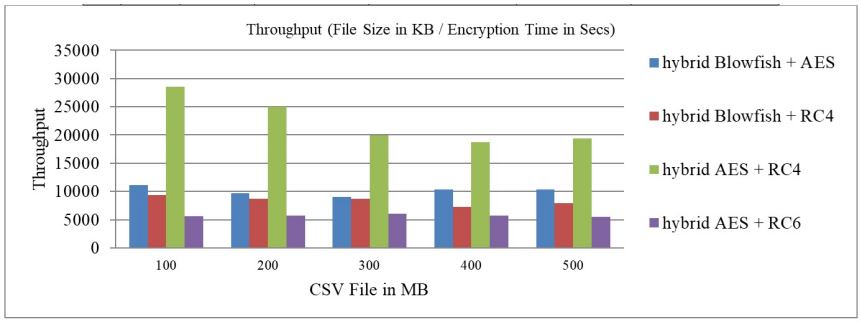

Fig. 18. Throughputs of Encryption for CSV Files

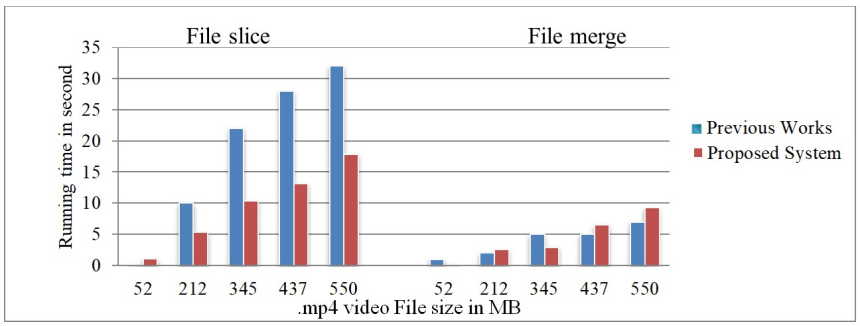

Fig. 19. File slice and merge time between existing and proposed system

Scenario 2: Running time comparisons of the proposed hybrid algorithm with the existing hybrid algorithm with file sizes ranging from 112 bytes to 153422 bytes are as follows.

Table 14. Running time comparison with the existing hybrid algorithms 1

	File Size in Byte	**Encryption Time in msec**		**Decryption Time in msec**	
		Existing Hybrid AES + Blowfish [26]	**Proposed Hybrid AES + RC4 with RSA**	**Existing Hybrid AES + Blowfish [26]**	**Proposed Hybrid AES + RC4 with RSA**
1	112	4389	69	3278	16
2	2305	5883	78	2634	23
3	7894	33900	79	12641	25
4	153422	194527	137	103314	90

Scenario 3: Running time comparisons of the proposed hybrid algorithm with the existing hybrid algorithms with file sizes ranging from 1MB to 4MB are as follows.

Table 15. Running time comparison with the existing hybrid algorithms 2

	Hybrid Algorithms	**Encryption Time in Sec**			
		1MB	**2MB**	**3MB**	**4MB**
1	Blowfish [25]	2.4	24.3	35.7	48.9
2	Blowfish & Krishna [31]	9.2	18.6	29.6	42.5
3	AES & Krishna [37]	3.4	12.3	23.7	36.3
4	Triple DES & Krishna [38]	3.1	7.1	13.4	18.1
5	AES & Blowfish & Krishna [39]	6.9	15.9	24.9	34.9
6	Proposed Hybrid Algorithm of AES and RC4 with RSA	0.422	0.526	0.652	0.991

Scenario 4: Running time comparisons of the proposed hybrid algorithm with the existing hybrid algorithm with file sizes ranging from 25 MB to 1024 MB are as follows.

Table 16. Running Time Comparison with Existing Hybrid Algorithm 3

	Hybrid Algorithms	**Encryption Time in Sec**				**Decryption Time in sec**			
		100	250	500	1024	100	250	500	1024
1	Existing hybrid Feistel, AES, HMAC [23]	25	50	75	105	25	50	75	105
2	Proposed AES + RC4 with RSA	3.59	16.79	26.49	62.49	3.95	13.5	29.61	58.77

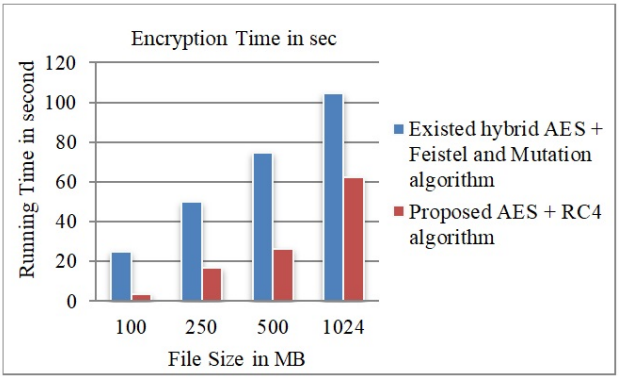

Fig. 20. Encryption time comparisons of existed and proposed algorithm

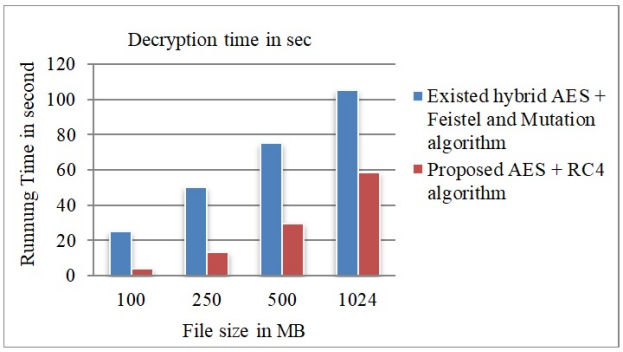

Fig. 21. Decryption time comparisons of existed and proposed algorithm

Table 17. Throughput comparison with the existed hybrid algorithm

	Hybrid Algorithms	**Throughput**			
		100MB	**250MB**	**500MB**	**1024MB**
1	Existing hybrid AES + mutation algorithm [23]	4000000	5000000	6666667	9752381
2	Proposed AES + RC4 with RSA	27870680	14888929	18873622	16386622

From the proposed hybrid algorithms, the hybrid AES + RC4 with RSA algorithm shows better performance in terms of running time and throughput than blowfish + AES with RSA, AES + RC6 with RSA, and RC4 + blowfish with

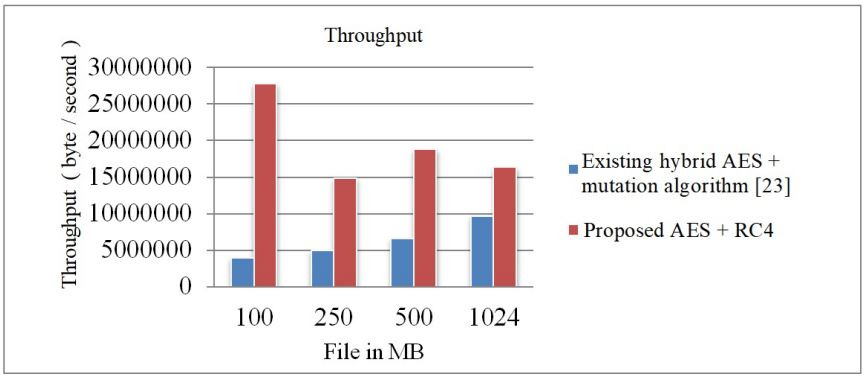

Fig. 22. Throughput comparisons of existing and proposed hybrid algorithm

RSA hybrid algorithms. Both the running (encryption and decryption) time and throughputs of the hybrid algorithm in [23] has lower performance than the proposed algorithm. Moreover, the proposed system keeps the secret key and hash value of the data at the local machine and sends only the encrypted data to the cloud. So, if the user wants to access the data, it downloads it from the cloud, and uses the secret key to decrypt the file and then uses the hash value to verify the integrity of the data. The secret key, the type of algorithm used for encryption and the hash value of the original data are kept private at the owner's machine to increase the degree of data security.

The peculiarity of this hybrid algorithm is the optimizing obtained due to the order of operations within each algorithms and the use of layered approach of the security algorithms. Moreover, this hybrid algorithms better speed and low resource consumption made is a better choice in legacy systems where the data is not highly sensitive RC4 can be used in combination with AES and RSA.

6. Conclusions and Future Works

6.1. *Contributions*

Enhance data security to store in a cloud storage system using a hybrid cryptography algorithm and proposed a better hybrid cryptographic algorithm in terms of running time, throughput, and degree of security. The proposed framework is implemented by integrating Microsoft Azure public cloud platform with the C# windows application. So, the cloud user sends only the encrypted data to the cloud server and the file information such as type of algorithm used, the private key, and hash value concerning the file names are stored in the user's local machine. This prevents both insider and outsider attacks from accessing the original data at transit or rest. The proposed system also avoids key leakage by encrypting symmetric keys with public-key cryptography. Analysis and construct a hybrid algorithm from the existing algorithms and proposed a hybrid algorithm that improves the

performance of data encryption processes and increases the degree of security of data.

6.2. *Conclusions*

The proposed system slices the file and encrypts the sliced parts and sends them to the cloud storage system. The proposed hybrid cryptography algorithms are presented in four ways which provide the end-user to select a suitable hybrid algorithm based on their data sensitivity. Both symmetric algorithms keys are encrypted via asymmetric algorithm and the encrypted keys are stored in the local machine to encrypted file parts. And the proposed system also avoids the problem of secret key leakage because the secret keys are encrypted with asymmetric keys. The proposed hybrid algorithm increases the level of securing data because data is encrypted with a different algorithm with different keys and those keys are also encrypted via an asymmetric algorithm to protect key leakage. From the four constructed hybrid algorithms, AES + RC4 hybrid algorithm with RSA is better because it takes lower running time for encryption and decryption of large files, and higher throughput. The results show that the proposed hybrid algorithm takes lower running time and the highest throughput than the existed hybrid algorithm.

6.3. *Future Works*

There are different cryptography algorithms and some of the algorithms need improvement to increase their strength. As future work, a new hybrid algorithm will be constructed from different existing algorithms to improve the performance in terms of running time, throughput, and the avalanche effect. So, the result of the newly constructed algorithm will be compared with our work and extend this algorithm by applying avalanche effect criteria. And, extend this work by integrating the proposed system with multiple public cloud platforms and including the backup and recovery of data and include mechanisms of tracking when accessing the data by insider and outsider threats.

References

1. Shubham Singh and Akhilendra Pratap Singh, "Ensuring Data Security in Cloud Storage ," *International Journal of Machine Learning and Computing*, vol. 8, no. 4, 2018 August.
2. Adamu Ismail Abdulkarim and Boukari Souley, "An Enhanced Cloud Based Security System Using RSA as Digital Signature and Image Steganography ," *International Journal of Scientific & Engineering Research* , vol. 8, no. 7, July 2017.
3. Sonia Rani and Harpreet Kaur, "Implementation and comparison of hybrid encryption model for secure network using AES and ElGamal ," *International Journal of Advanced Research in Computer Science* , vol. 8, no. 3, April 2017.

4. S.V.N.Srivalli and Ben Swarup Medikonda, "Development of a Cloud-based Secure Text File Application using Hybrid Cryptography and Steganography ," *International Journal of Recent Technology and Engineering* , vol. 8, no. 1, may 2019.
5. Bhushan Rathod and Prashant Yelmar, "Efficient Cloud Security Method for Preventing Insider Attacks in Cloud Computing Platforms ," *International Journal of Advanced Research in Computer and Communication Engineering* , vol. 6, no. 6, June 2017.
6. Ali Abdulridha Taha et al., "Developing New Hybrid Cryptography Algorithm for Cloud Computing Environment," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 11, 2017.
7. Vimala.M and SriPreethaa.K.R, "Multilevel data security protection mechanism for cloud storage system," *International Journal of Science, Engineering and Technology Research*, vol. 6, no. 5, May 2017.
8. Mihir Shah, "Hybrid cryptosystem for secure data storage," *International Journal of Innovative Research in Information Security*, vol. 4, no. 11, November 2017.
9. Ponnuru Sowjanya and K. V. N. Sunitha, "Enhancing the data security in cloud using hybrid algorithm," *International Journal of Pure and Applied Mathematics*, vol. 120, no. 8, June 2018.
10. Shilpi Harnal and R.K. Chauhan, "Hybrid Cryptography based E2EE for Integrity & Confidentiality in Multimedia Cloud Computing," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 10, August 2019.
11. Diaa Salama AbdElminaam, "Improving the Security of Cloud Computing by Building New Hybrid Cryptography Algorithms," *I.J. of Electronics and Information Engineering*, vol. 8, no. 1, March 2018.
12. Shakeeba S. Khan, R. R. Tuteja, "Data Security in Cloud Computing Using Cryptographic Algorithms," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 7, no. 1, January 2019.
13. Dhuratë Hyseni and Artan Luma, "The Proposed Model to Increase Security of Sensitive Data in Cloud Computing," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 2, 2018.
14. Aditya Poduval et al., "Secure File Storage on Cloud using Hybrid Cryptography," *International Journal of Computer Sciences and Engineering*, vol. 7, no. 1, January 2019.
15. K. Subramanian and F.Leo John, "Enhanced Security for Data Sharing in Multi Cloud Storage," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 3, 2017.
16. P.Chinnasamy and P.Deepalakshmi, "Design of Secure Storage for Health-care Cloud using Hybrid Cryptography," in *Proceedings of the 2nd International Conference on Inventive Communication and Computational Technologies* , Srivilliputtur, 2018.

17. Neha and Mandeep Kaur, "Enhanced Security using Hybrid Encryption Algorithm," *International Journal of Innovative Research in Computer and Communication Engineering* , vol. 4, no. 7, July 2016.
18. Nadesh R.K et al., "Enhancing security for end users in cloud computing environment using hybrid encryption technique", *International Journal of Engineering & Technology,* 7 (1) (2018) 152-156
19. V. Kapoor and Rahul Yadav, "A Hybrid Cryptography Technique for Improving Network Security ," *International Journal of Computer Applications* , vol. 141, no. 11, May 2016.
20. Munavvara Tahaseen et al. , "Data Storage on Cloud Using Hybrid Encryption with One Time Password ," *IOSR Journal of Computer Engineering*, vol. 19, no. 4, August 2017.
21. Joseph Selvanayagam et al., "Secure File Storage On Cloud Using Cryptography," *International Research Journal of Engineering and Technology* , vol. 5, no. 3, March 2018.
22. Rohit Barvekar et al., "An Approach To Hybrid Cryptography On Cloud Environment," *IJARIIE-ISSN(O)-2395-4396* , vol. 4, no. 2, 2018.
23. Ahmad Habboush, "Multi-Level Encryption Framework," *International Journal of Advanced Computer Science and* Applications, vol. 9, no. 4, September 2018.
24. Mohammed Nazeh Abdul Wahid et al., "A Comparison of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish for Guessing Attacks Prevention," *Journal of Computer Science Applications and Information Technology* , August 2018.
25. Theda Flare G. Quilala et al., "Modified Blowfish Algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 12, no. 1, October 2018.
26. Hossein Abroshan, "A Hybrid Encryption Solution to Improve Cloud Computing Security using Symmetric and Asymmetric Cryptography Algorithms", *(IJACSA) International Journal of Advanced Computer Science and Applications,* Vol. 12, No. 6, 2021
27. T. Nie, C. Song and X. Zhi, "Performance Evaluation of DES and Blowfish Algorithms," *2010 International Conference on Biomedical Engineering and Computer Science,* 2010, pp. 1-4, doi: 10.1109/ICBECS.2010.5462398.
28. Salim Ali Abbas et al., "Enhancing Security of Cloud Computing by using RC6 Encryption Algorithm," International Journal of Applied Information Systems (IJAIS), vol. 12, no. 8, November 2017.
29. Mustafa S. Abbas, Suadad S. Mahdi, and Shahad A. Hussien, "Security Improvement of Cloud Data Using Hybrid Cryptography and Steganography", 2020 International Conference on Computer Science and Software Engineering (CSASE), Duhok, Kurdistan Region – Iraq, 978-1-7281-5249-3/20/$31.00 ©2020 IEEE
30. Fursan Thabit et al., "A new lightweight cryptographic algorithm for enhanc-

ing data security in cloud computing", *International Journal of Intelligent Networks* 2 (2021) 18–33

31. Anjali D.V, and S.N Chandrashekara, Design and Implementation of Secure Cloud Storage System using Hybrid Cryptography Algorithms with Role based Access Control Model, *International Journal of Engineering and Technical Research (IJETR)* ISSN: 2321-0869 (O) 2454-4698 (P), Volume-5, Issue-1, May 2016
32. Jindal P, Singh B. Performance analysis of modified RC4 encryption algorithm. InInternational conference on recent advances and innovations in engineering (ICRAIE-2014) 2014 May 9 (pp. 1-5). IEEE.
33. Lei D. F-HASH: Securing Hash Functions Using Feistel Chaining. IACR Cryptol. ePrint Arch.. 2005;2005:430.
34. Shivaramakrishna D, Nagaratna M., "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. 2023 Dec 1;84:275-84.
35. Salman, Doaa S., and Jolan Rokan Naif. "Smart Cloud Security Using Hybrid Encryption Algorithms With 4-D Chaotic Key." *Journal of Research Administration* 5, no. 2, 2023, pp. 1816-1837.
36. Abroshan H., "A hybrid encryption solution to improve cloud computing security using symmetric and asymmetric cryptography algorithms." *International Journal of Advanced Computer Science and Applications*, *12* (6), 2021, pp.31-37.
37. Mageshkumar N, Swapna J, Pandiaraj A, Rajakumar R, Krichen M, Ravi V.,"Hybrid cloud storage system with enhanced multilayer cryptosystem for secure deduplication in cloud." International Journal of Intelligent Networks. 2023 Jan, 1 (4), pp. 301-309.
38. Pothireddy S, Peddisetty N, Yellamma P, Botta G, Gottipati KN.. "Data Security in Cloud Environment by Using Hybrid Encryption Technique: A Comprehensive Study on Enhancing Confidentiality and Reliability.", International Journal of Intelligent Engineering & Systems. (2024) 17(2).
39. Singh, A., 2024. Design of An Effective Hybrid Cryptographic Technique For Information Security In Cloud Computing. *Migration Letters*, *21*(S5), pp.493-508.