AKADÉMIAI KIADÓ

UNIVERSITY of DEBRECEN

International Review of Applied Sciences and Engineering

13 (2022) 2, 107–116

DOI:
10.1556/1848.2021.00317

ORIGINAL RESEARCH PAPER

A novel data classification-based scheme for cloud data security using various cryptographic algorithms

Mohd Naved Ul Haq[1] and Narender Kumar[2*]

[1] Department of Computer Science and Engineering, H.N.B. Garhwal University, Srinagar Garhwal, India

[2] Department of Computer Science, Doon University, Dehradun, India

Received: May 28, 2021 • Accepted: August 3, 2021
Published online: September 23, 2021

ABSTRACT

Digital technologies had an effect on people's lives. The majority of these digital devices rely on cloud storage to meet their memory needs. Hundreds of thousands of images, videos, and audio files are being transferred to cloud storage. Thousands of people around the world access these media every second. Unauthorized access to these media must be avoided. One of the weak points for data breaches is the user-end encryption. This paper suggests a strategy for improving cloud data protection by combining the AES and blowfish encryption and decryption algorithms. AES-256 is used as the first layer, followed by blowfish as the second layer, in the hybrid solution. The output of the first layer is input to the second layer and the final result is analyzed. The proposed method also discusses other combined approaches such as AES with other traditional algorithms but the proposed method gives significant results compared to other approaches.

INDEX TERMS

cloud computing, data security, cloud storage, client-side encryption, AES, DES, 3DES, blowfish

1. INTRODUCTION

With the widespread adoption of cloud computing in our daily lives, the cloud has evolved into a data storage platform. One of the most significant barriers to cloud adoption, particularly among corporate users, is data security. One of these topics is cloud security, which includes things like technology, control, and a set of regulations that are beneficial in safeguarding the relevant data. Apart from these services, cloud security also addresses the numerous risks and assaults that may have an indirect or direct impact on a system.

Cloud security encompasses issues such as data integrity, source availability, and confidentiality of the cloud infrastructure as a whole, as well as specific layers and their services [1]. This brings up another point: encryption and decryption are resource-intensive operations that take up a lot of CPU, memory, and time.

Another issue to keep in mind is that not all data is equal. Some data must be kept private, but others, such as material in public training manuals, advertisement media, and so on, can be left unencrypted without causing any difficulties for end-users.

The remaining data is likewise not uniform in terms of the level of protection required. It may be further separated and encrypted using several degrees of encryption, with stronger encryption reserved for more sensitive data. Although a single strong cypher like AES may be regarded sufficient for data security, there is still a theoretical worry about trusting components like AES′ static S-Box. To address this problem, several cyphers in the chain have been recommended to minimize the possibilities of the secret data being compromised [40].

*Corresponding author.
E-mail: narenrawal@gmail.com

AKJournals

Virtualization, multi-tenancy cloud storage, and a cloud network are the core cloud computing components [3,4,5,6], which is why we suggest this method. The three service models used by a cloud service provider to deliver cloud services to its customers are Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), and Infrastructure-as-a-Service (IaaS) [7]. Four types of cloud deployment models (Public, Private, Hybrid, and Community) that meet the essential business demands of cloud users [7]. The following are the categories in which we have categorized cloud security issues: computing security, data storage, virtualization security, internet services-related security, network security, access control, software security, trust management, and legal security and compliance problems [2]. The National Institute of Standards and Technology provides a list of criteria that are required for cloud security. In its dissemination of regulations for cloud security standards, the CSA has emphasized the same point. Authentication, confidentiality, integrity, and availability are among them [8]. After determining the execution time of all the methods, we merged each algorithm with AES 256 to calculate the execution time of all the encryption and decryption.

We attempted to compare all of these findings to AES 256 and determine which method had the fastest execution time. In the suggested scheme, AES 256 and Blowfish have been assigned to Level 3. This is because AES 256 and Blowfish have the fastest execution times, and the algorithms covered are AES encryption and decryption, as well as AES-Blowfish cascade encryption and decryption.

2. LITERATURE SURVEY

Kumar et al. [9] provided a comprehensive examination of all aspects of cloud security. The poll began with an in-depth explanation of the cloud computing idea. They then suggested a cloud computing security taxonomy, which was groundbreaking in its own right. The newest literature was used to break down and expose vulnerabilities in cloud security in detail. After that, the topic of cloud security countermeasures was covered in great depth. They conducted a comprehensive examination of all of the issues discussed.

The article concluded with a discussion of cloud security in the context of future technologies such as 5G and Big Data. [10] The study also discussed the cloud computing model's different stakeholders, such as the cloud provider, broker, auditor, consumer, and carrier.

The authors presented CHiS-256, a symmetric block cypher, in [11]. They have proposed this as a method for securing cloud data. When compared to the RC5 cypher, the suggested encryption had superior normalized performance. The authors also proposed a method for storing data in a safe manner utilizing metadata. The authors of [12] compared several cryptographic algorithms, cryptographic data structures, Hash processes, digital signatures, and other cryptographic methods. Finally, the suggested architecture ensures data availability, confidentiality, privacy, and availability, among other things.

Singh and Thokchom [13] proposed a method for ensuring the integrity of data saved in cloud settings without having to retrieve the full data set. Identity-based ring signature, vector commitment, and group-based key agreement protocols were used in the approach they presented. In [14], the authors described a work-in-progress method that involves the deployment of a proxy server to encrypt data before distributing it to multiple cloud providers for storage.

Ramchandra et al. [15] also covered the restrictions, weaknesses, and dangers involved with cloud computing in general, as well as the remedies available.

Singh et al. [16] conducted a thorough review of the state of the art in the field of cloud security. Then they created a new taxonomy of cloud computing security concerns, including embedded security, client administration, data storage, web applications, metadata, and so on.

Finally, they made cloud security suggestions based on several classification techniques, such as infrastructure, service middleware, and application layers. The authors conducted a survey focusing on cloud security issues in [17]. The threat from many directions was then explored. Coppolino et al. [18] conducted a study that uncovered a wealth of information on data threats, network threats, and other risks associated with cloud systems. Network, hardware, and hypervisor are the vectors they discuss.

Then they talk about how to counteract the various attack vectors and how to break them down into subfields. [2] begins by comparing previous work on cloud security, trust, and solutions, among other topics. The articles were compared using a matrix of the many categories they address, such as cloud trust, security needs, open issues, and solutions, among others.

They also went through the CSA's cloud security risks in great depth. They spoke about DoS attacks, metadata spoofing, and the user to root attack when it came to assaults. They divided the problem into eight subcategories under cloud security concerns, then further divided each subtype into many kinds. Trust, service availability, and data recycling were among the topics covered under the category of data storage, which was very significant to us. The study came to a close with a consideration of research gaps. The authors of [19] focused their discussion on methods for protecting data stored in the cloud. Authentication and authorization techniques, as well as their problems, were explored in depth in the study. Another important topic was access methods, including a comparison of McAfee and Fujitsu's solutions. The author of [20] covered the fundamentals of cloud threats before moving on to a case study of cloud security in the Amazon Web Service. Physical security, accreditations, and certifications were all included in the case study.

It also explained how Amazon manages numerous users using their access control system. The approach used features including storage, content, and access control in [21]. They then presented an example of data elements relating to a person, and utilized their categorization method to assign various classification qualities to it. Although their system provided different attributes over which to categorize the

data, one drawback of the study was that it did not provide a means to employ all or multiple attributes to offer some category in which to classify the object in issue. The system also used the notion of dividing data into numerous sections before encryption and subsequently encrypting them, as described in [22].

This was done to make the encryption process take longer. As a result, while attempting to retrieve the plaintext files, an attacker would have to use more computing resources. For increasing security levels, they used AES 128, 256, and 512. In [23], the authors presented a concept for leveraging clouds to provide encryption as a service, commonly known as the EaaS paradigm. They devised a system for assessing the service's quality based on four factors.

The CRITIC method for quality quantification was also given a quality measure, as well as factors for weighting it. The authors of [24] proposed a method for homomorphic encryption of data in the cloud to allow users to use potentially untrustworthy cloud providers.

The suggested method is capable of both additive and multiplicative homomorphism. To make things clearer, the proposed technique employed a probability-based approach and was also illustrated with an example.

The authors presented a technique in [25] for conducting accounting operations on data saved in the cloud while the data is encrypted and hence useless to an attacker. After being encrypted with homomorphic encryption, the data was saved in AWS′ DynamoDB. The homomorphic encryption enabled addition and subtraction operations on the encrypted data. The method proved that arithmetic operations could be performed on encrypted data in the cloud. The authors suggested a modification to PKEET-FA, a flexible authorization-based public-key encryption method, in [26]. The authors' technique enabled them to validate the cloud's honesty during the equality test. For 5G and cloud computing security, the recommended solution is crucial. The authors of [27] proposed a novel method for assessing a cloud provider's cloud security using business process modelling. The article began by giving an outline of cloud and data security. After that, a brief outline of business process modelling was given. The strengths and drawbacks of three cloud service providers' business process modelling of their cloud security strategy were discussed.

The cloud providers were de-identified and not mentioned. In a hierarchy-based procedure, the authors developed a unique attribute-based encryption technique for file attributes in [28]. The access structures, hierarchical access trees, bilinear map, and DBDH assumption were all concepts employed in the method. Multiple keys are created depending on the access tree in this method. The data is subsequently saved in the cloud and may only be accessed by entities that meet some or all of the access tree's requirements.

As a result, the owner was able to ensure that the data could be accessible by a large number of members of organizations based solely on their identification attribute, thus enabling hierarchical access control.

The authors of [29] presented a method for dealing with user and cloud provider trust concerns. By analyzing the user's previous behavior, the cloud provider tries to quantify the user's trustworthiness. After this study, if the user looks to be dangerous, they are branded as such, and the cloud provider can opt to limit their access. The attribution of trust evidence and characteristics is necessary for this computation.

The authors tested their method on a genuine cloud platform, generating trust levels for a range of users.

Yan et al. [30] published an editorial that summarized the state of the art in the field of cloud data security and cryptography. They began by discussing the most recent research in the field of cloud data storage security. The suggested works included methods for cloud-based hierarchical key assignment, attribute-based encryption for video-on-demand services, and an intelligent technique for keeping partial data safe from cloud providers.

Another intriguing research paper suggested a generic number field sieve method capable of parallel execution to make use of cloud computing's parallel nature. The authors suggested and verified an attribute-based encryption method that was also searchable in [31]. This technique was discovered to be intriguing because looking through encrypted material is generally not possible without full decryption. Plain data must be encrypted and uploaded to the cloud by the data owner. They also submit an encrypted data index.

The system entailed the use of a trusted authority to distribute private keys to end-users who needed to search the data for a certain keyword.

The end-user would then use the secret key as a backdoor to search the encrypted index for the keyword. The proposed technique was also subjected to a theoretical and experimental investigation in the publication. The authors suggested a method for securely storing data in a public cloud in [32].

Encryption was used first, followed by steganography to store the data in the form of images. The encrypted data was stored in the image file's unused bits as part of the steganography procedure.

DWT steganography was utilized as the steganography technology.

RSA was the encryption method utilized. The usage of two keys, public and private keys, was required for this public key cryptography technique. The procedure produced a stego file that appeared identical to the cover file, ensuring that the data's presence remained secret. The challenge of maintaining the secrecy of data undergoing data mining in a cloud context is discussed in [33]. The challenge was to ensure that given n sites containing the data being mined for patterns, I would not be able to determine the patterns discovered from the site. The problem, according to the paper's literature assessment, was one that had received a lot of attention.

The authors of [34] offered a theoretical examination of the Selective Opening assault scenario. This problem occurs in cloud environments when many users utilize the cloud

provider's public key to encrypt data sent to the cloud. When an attacker has access to certain ciphertext and can alter it, this is called a ciphertext attack. The author of [35] provides an overview of the different methods that may be used to encrypt data in cloud settings. They divided their research into two categories: symmetric and asymmetric methods.

Data encryption standard (DES), advanced encryption standard (AES), blowfish, triple DES, and the international data encryption method were among the symmetric algorithms explored (IDEA). The Rivest Shamir Adelman (RSA) method and Diffie Helman key exchange were among the asymmetric encryption techniques presented. The authors of [36] provide a comprehensive overview of the many cryptographic techniques that can be used. The survey started with an overview of cloud computing features provided by the National Institute of Standards and Technology.

Following that, the requirements, risks, and weaknesses of cloud security were discussed. Data secrecy schemes, such as attribute-based, homomorphic encryption, were thoroughly discussed. Remote data integrity checks and data privacy protection were also explained in the same way. Overall, the article provided a very thorough and interesting overview of the cryptographic solutions for cloud security that are now available [37] proposes a method that combines steganography and public-key encryption. This technique uses a mediated certificateless encryption procedure that utilizes a cloud-based key distribution hub. On the cloud, too, is the mediator. The author provided a survey of data security concerns in Cloud security in [38].

The author began with an overview of the cloud service paradigm before moving on to the variables that influence cloud security. These issues were dealt with individually. The authors of [39] devised a way for encrypting video using Hadoop and AES over the cloud. Allocating the duty of encrypting videos to parallel task servers is done using the MapReduce architecture. As a result, the method may take use of cloud computing's parallel nature to complete a resource-intensive operation like encryption in less time than previous techniques. The authors of [41] conducted a theoretical investigation of homomorphic encryption in the context of known plain text attacks.

Complete, partial, and selective encryption options are also available with this method. To improve the privacy of data kept on the cloud, the authors used a hybrid encryption method in [40]. The encryption was performed with the help of homographic and blowfish techniques. They also offered a rundown of the many options available.

3. PROPOSED WORK

The suggested classification divides the information into three categories. The user is expected to decide which data belongs to which category intuitively [42].

The solutions offered to address the problem of cloud data security include the adoption of a categorization system for data stored in the cloud. The data may be kept without client-side encryption or encrypted using one of two encryption methods, depending on the categorization. As illustrated in Fig. 1, the system is separated into three phases: client-side encryption, cloud storage, and client-side decryption.

The three stages are as follows:

1. Client-side encryption phase

As illustrated in Fig. 2, this may be further split into three categories based on the relative relevance of the data. The client-side choice regarding the data category is made by the end-user and is not described in the scheme.

Level 1: This is non-sensitive information that may be made public. This includes public CVs, public keys, and material that has previously been made public, among other things. This data is not encrypted on the client side and is sent to a cloud provider in its unencrypted form.
Level 2: This is data that the customer considers to be of medium importance and so requires encryption while being stored on the cloud platform.

Family photographs, personal papers, and other sorts of data may be included in this category.

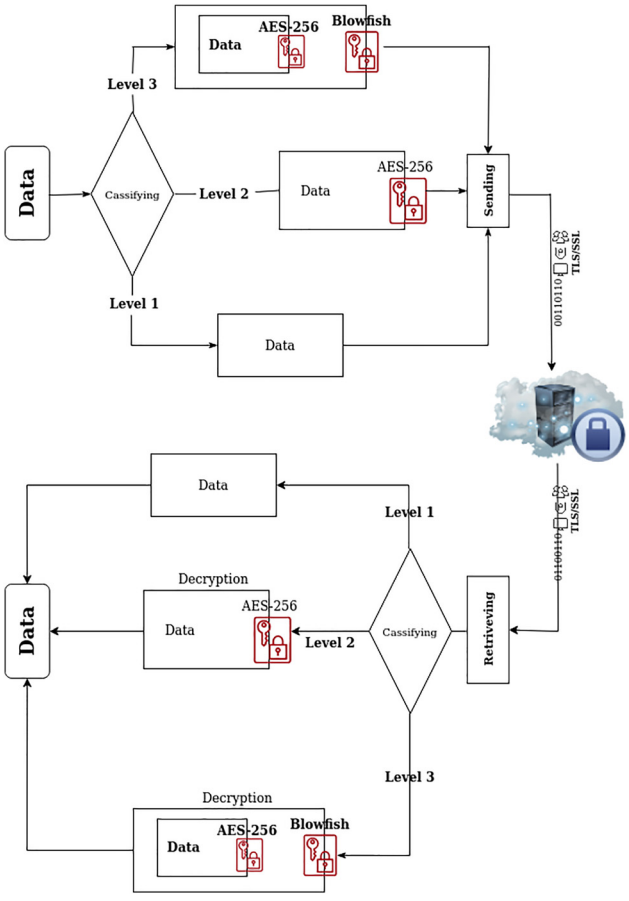

Fig. 1. Schematic diagram of the proposed scheme. Ciphers: AES-256 and Blowfish

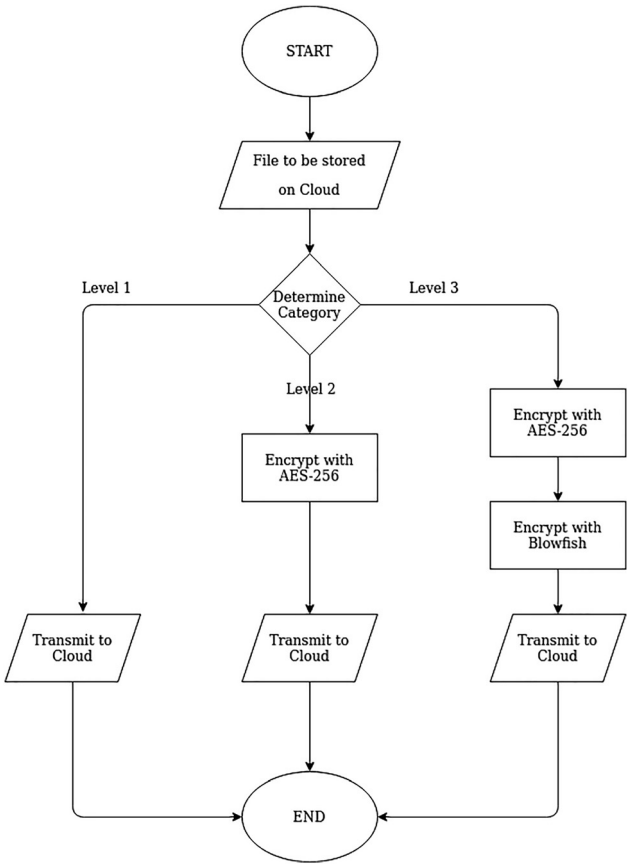

Fig. 2. Flowchart for the encryption scheme

Before being transferred to the cloud provider, this data is encrypted on the client side using AES-256 encryption. The encryption's private key is kept on the user's computer.

Level 3: This is the data that is regarded to be of the utmost significance, to the point that the theoretical scenario of a single cypher failing becomes relevant, and the user need the highest level of protection for their data. This entails the use of a cascade of two ciphers. The data is initially encrypted with one key using AES-256, and then the output is further encrypted with a second key using the blowfish cypher.

Both keys are kept on the user's person.

2. Cloud storage phase

This is the phase that involves data transmission between the client and the cloud, as well as data storage on the cloud platform. There may be an HTTPS connection between the client and the cloud, if the cloud provider supports it. The cloud provider may encrypt the data further once it is present, but our technique does not rely on this situation because pertinent data is already encrypted using one of two methods before transmission.

3. Client-side decryption phase

Based on the type of data in question, this phase can be split into three groups.

Level 1: Because this data was saved without client-side encryption, it may be retrieved without the user having to do any decryption procedures.
Level 2: This data is encrypted with AES-256, and the user has access to the key.
Level 3: This data must be decrypted first with Blowfish and then with AES-256 using the two keys used for encryption in the first step. The National Bureau of Standards, subsequently known as the National Institute of Standards and Technology, authenticated DES in 1977. The DES algorithm takes 64 bits of input text, such as plain data, and produces 64 bits of ciphertext data as an output. Each 64-bit block of input is encrypted in 16 cycles using the algorithm. This algorithm's symmetric key size is 56 bits, while the block size is 64 bits [45]. When compared to DES, the 3DES algorithm encrypts twice as much. As a result, the encryption quality improves and becomes more difficult to crack.

A triple DES is a block cypher that uses a key that is 168 bits long and works on 48 rounds (three times SDS) in various counts. During encryption, the method 3-DES uses a block size of 64 bits [46]. Instead of employing Feistel networks like many other cyphers, the AES employs substitution-permutation networks.

The substitution-permutation approach employs substitution and permutation boxes to efficiently spread plaintext information across the ciphertext.

It is based on Joan Daemen's and Vincent Rijmen's Rijndael cypher [43]. Bruce Schneier proposed the blowfish cypher algorithm. The algorithm's main goal was to create a totally open structure with the extra benefit of dynamic S-Boxes. Rather than being defined statically by the Blowfish implementation provider, these S-Boxes are key-dependent [40].

This protects the S-Box, the algorithm's major driving force. Both of the cyphers in our system are block cyphers, meaning they only function on a single block of data at a time. The cypher block chaining mode [44] was the block cypher mode of operation we used in our method. Before encryption, the initialization vector (IV) of the first block of data is XORed in this mode.

IV is a block-sized random collection of data.

The output is encrypted using the appropriate cypher after the first block is XORed with IV. The encryption result is then XORed with the next block to be encrypted. In CBC mode, decryption is the inverse of encryption in that the XOR operation occurs after the decryption. The ciphertext blocks are decrypted before being XORed with the ciphertext block before it.

The proposed study will concentrate on the client-side cryptographic services offered by the proposed system.

It stresses the suggested scheme's implementation of the Level 2 and Level 3 categories' capabilities. According to the table below, the AES 256 and blowfish cascade cyphers are the safest and fastest. There has not been an assault on either of these algorithms yet. Because these two are so safe and quick, the work suggests using both of them. All other algorithms, according to a literature analysis, are not as secure

and quick. Some academics claim that there is an attack on AES 256, however the majority of researchers claim that the algorithm is 99 percent secure thus far.

3.1. Hardware

The proposed work used intel core i3 of 7th generation, CPU speed Dual-core 2.30 GHz, OS windows 10, RAM 8 GB, SSD 120 GB which read/write speed is 545 Mb s^{-1}. The work is experimented on eclipse IDE using Java programming.

3.2. Algorithm explanation

Algorithm 1: AES-256 encryption

1. Key generation
2. Pad the input file to make it a multiple of the block size of AES.
3. Create IV for CBC mode
4. Place IV at the beginning of the output file
5. Read blocks of the file.
6. Encrypt block read in step 4
7. Add encrypted block from step 5 to the output file
8. While more blocks remain, go to step 4

Key generation in this work is implementation IV uses the object of the "IvParameterSpec", a class is performed to generate a random byte stream of a given size and then putting it through SHA-256 to generate a usable 256-bit key. Since the cipher is using the CBC mode of operation, so the padding of inputs and generating the IV are the next steps that are taken. The IV is saved with the output file as head of the output file to use it during decryption. Then a loop is operated, which keeps removing and encrypting one block of data at a time from the padded input file. The loop continues until blocks are remaining to be read from the input file.

Algorithm 2: AES-256 decryption

1. Load prestored key
2. Read the first block of encrypted file to read the IV
3. Read blocks of the file
4. Decrypt block read in step 3
5. Write the output of step 4 to the output file.
6. While blocks remain in the input file got to step 3
7. Unpad output file to obtain the original file.

The decryption process requires the loading of the key that was generated during the process of AES-256 encryption. Then the IV is recovered by reading the first block of the encrypted file. Then a loop starts, which mimics the operation of the loop, used in the encryption process. The loop reads successive blocks of data from the encrypted file and decrypts them, and places them in an output file. After the decryption process is completed, the generated data is unpadded.

Algorithm 3: Blowfish encryption

1. Generate Blowfish key
2. Pad input data to make it multiple of block size
3. Create initialization vector
4. Generate S-Boxes
5. Generate Subkey array
6. Perform block by block encryption with CBC mode
7. Write output to the output file

Blowfish key is generated separately from the AES-256 encryption since reusing the key between the two ciphers in the cascade mode of operation would result in the defeat of the entire purpose of using a cascade cipher. Blowfish is using the CBC mode. It is required to ensure that the data being encrypted is a multiple of the block size of the blowfish cipher, which is 64 bits. Thus, after the key is generated, the next step is to pad the input data to the nearest 64 bits. The padding technique that has been used here is the PKCS#5 padding scheme. In this scheme, the value of the padded bits is equal to the number of the pad bits that are being added. This allows for easy identification of the padded portion. The PKCS#5 scheme also adds an entire block of pad bits if input data is already a multiple of the block size. The next step is the generation of the IV since the encryption mode here is CBC, hence IV is mandatory to ensure the strength of the block chaining mode of operation. The IV is generated using the object of the "IvParameterSpec" class. The S-boxes have been generated automatically as a part of the process of blowfish's operation. Similarly, the subkey array of 18 subkeys is also generated. Finally, the block-by-block encryption of data can proceed using the statement:

```
IvParameterSpec iv = new IvParameterSpec (IV.getBytes (StandardCharsets.UTF_8));

cipher = Cipher.getInstance("Blowfish/CBC/PKCS5Padding");

Secretes secretKey = new SecretKeySpec (KeyData, "Blowfish");

cipher.init(Cipher.ENCRYPT_MODE, secretKey, iv);
```

Algorithm 4: Blowfish decryption

1. Load key and IV
2. Read the input file.
3. Perform block by block decryption under CBC mode
4. Unpad the output
5. Write the unpadded output to the output file

The blowfish decryption process is the reverse of the blowfish encryption process. Since the decryption is performed in CBC mode, so the decryption also needs to be performed in CBC mode. For this, the original IV is needed. Unlike the case of AES, where the IV was stored with the encrypted file itself in the case of blowfish, the IV was not stored with the encrypted file. It is stored with the user, the same as the key. The decryption process begins with the loading of both the IV and the key. They are passed together as arguments to the CBC mode decryption command:

```
IvParameterSpec iv = new IvParameterSpec (IV.getBytes (StandardCharsets.UTF_8));
```

cipher = Cipher.getInstance("Blowfish/CBC/PKCS5Padding");

Secretes secretKey = new SecretKeySpec (KeyData, "Blowfish");

cipher.init(Cipher.DECRYPT_MODE, secretKey, iv);

This statement is the heart of the blowfish decryptions process, which functionality is provided by the blowfish java library. The key is stored in the cipher object in a previous step. The output obtained after the decryption still needs the padding added during the encryption to be removed. The next statement removes the PKCS#5 padding that was added during the encryption step. The data obtained is the original file, which is then written to a file and put to use.

4. RESULTS & DISCUSSION

The discussion provides experimental results of the approach. The results are accompanied by a detailed evaluation to better understand the inference provided by the results. The experiments were performed on files of various sizes such that they reasonably covered files of various sizes. To better analyze the results, files are treated in two categories based on their sizes. According to our approach to encryption and decryption, the proposed work uses four algorithms and their pairs such as AES-256 +DES, AES-256+3DES, AES-256+Blowfish cascade cipher. So, all of these cannot be reasonably compared to each other. Therefore, they are all analyzed separately. For a given file of a size, we performed the encryption and decryption procedures for five iterations to get a more accurate average value. The features of both blowfish and AES 265. The increasing importance of a safe internet for a user and securing data storage is the need of an era. If analysis is performed on a multitasking system, there might be variations in each iteration given the varying load states of the CPU. Initial experiment analyses on DES, 3DES, AES256, and blowfish cascade ciphers and took two types of files, the first ranging from 1 to 5 MB [Fig. 3] and the other from 100 KB to 500 KB [Fig. 4]. After that we analyzed on AES-256 +DES, AES-256+3DES, AES-256+Blowfish cascade cipher and similarly for that also we have taken file from 1 to 5 MB [Fig. 5] and 100 to 500 KB [Fig. 6]. As given in the

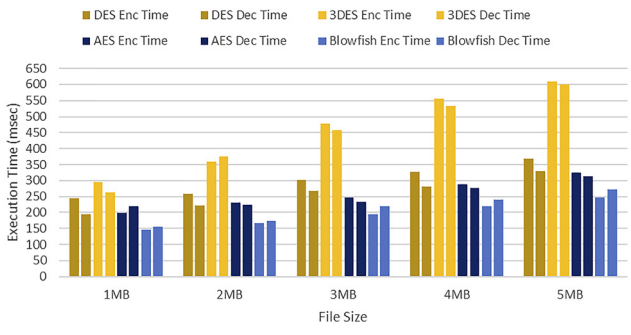

Fig. 3. Encryption and Decryption graph for 1–5 MB file size

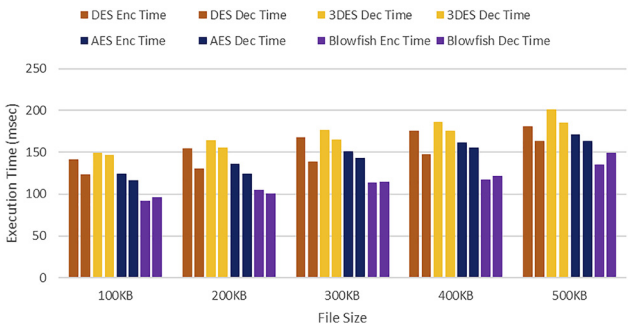

Fig. 4. Encryption and Decryption graph for 100 to 500 KB file size

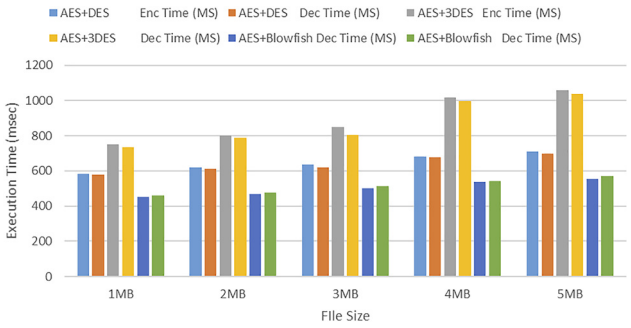

Fig. 5. Encryption and Decryption graph for 1–5 MB file size

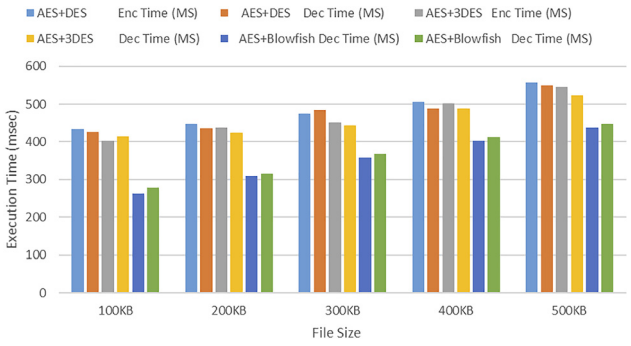

Fig. 6. Encryption and Decryption graph for 100 to 500 KB file size

table, the obtained execution time of both file sizes. Therefore, one thing immediately clear from the table is that execution time for small files appears to be more or less linear. This fact can be further inferred from the graph of the tables below, which shows that encryption and decryption increase the execution time linearly which are given in the table and graph below (Tables 1–5).

4.1. In both of these tables, individual are the results of algorithms (DES, 3DES, AES, blowfish)

4.2. Below are the results of paired algorithms in both of these (AES+DES, AES+3DES, AES+Blowfish)

The above analysis shows execution time of various algorithms and further shows execution time of their pair algorithms.

Table 1. Execution time (millisecond) of encryption, decryption for file size 1 to 5 MB

File Size (MB)	DES Enc Time	DES Dec Time	3DES Enc Time	3DES Dec Time	AES Enc Time	AES Dec Time	Blowfish Enc Time	Blowfish Dec Time
1 MB	244.17462	193.1063	295.09574	263.14428	198.56256	218.7699	146.89107	154.10026
2 MB	257.68414	221.25332	360.00884	374.79086	231.12706	225.12332	166.36624	172.75584
3 MB	301.55536	268.26512	478.8577	457.87246	247.63216	232.93404	193.25002	219.23742
4 MB	328.21974	281.73858	557.00402	534.06456	287.99504	276.65112	218.56203	239.02434
5 MB	369.4479	328.67228	609.47772	598.7873	324.11608	314.13816	247.79102	271.71302

Table 2. Execution time (millisecond) of encryption, decryption for file size 100 to 500 KB

File Size (KB)	DES Enc Time	DES Dec Time	3DES Dec Time	3DES Dec Time	AES Enc Time	AES Dec Time	Blowfish Enc Time	Blowfish Dec Time
100 KB	141.59958	123.3447	149.0343	146.21262	124.86542	116.19916	91.97014	96.64504
200 KB	154.04642	130.7884	163.91364	155.05628	136.20364	124.59802	104.95352	100.62482
300 KB	167.25358	138.67356	176.15386	164.88298	150.61282	142.74378	113.90728	114.54082
400 KB	175.94922	147.09262	185.85314	175.78762	161.12756	155.18706	117.30498	121.8407
500 KB	180.66342	162.91238	201.4228	184.97078	171.43706	163.25962	134.95322	148.8612

Table 3. File Size 1–5 MB

File Size (KB)	AES+DES Enc Time (MS)	AES+DES Dec Time (MS)	AES+3DES Enc Time (MS)	AES+3DES Dec Time (MS)	AES+Blowfish Dec Time (MS)	AES+Blowfish Dec Time (MS)
1 MB	584.08092	580.05138	750.2494	735.71238	451.88728	461.43066
2 MB	619.89276	609.75568	801.52258	786.12808	469.20764	474.59464
3 MB	635.34606	621.0125	851.11958	804.48118	501.06008	511.51182
4 MB	681.25662	676.78	1,017.86576	997.46862	535.89818	541.89996
5 MB	711.38526	696.2091	1,060.37354	1,038.11418	552.01552	568.57898

Table 4. File size 100 to 500 KB

File Size (KB)	AES+DES Enc Time (MS)	AES+DES Dec Time (MS)	AES+3DES Enc Time (MS)	AES+3DES Dec Time (MS)	AES+Blowfish Dec Time (MS)	AES+Blowfish Dec Time (MS)
100 KB	433.26462	425.89958	403.22024	413.74392	263.45406	278.54268
200 KB	446.80528	436.61669	437.20894	423.30882	309.09582	314.61348
300 KB	475.48406	483.961469	451.64358	444.17338	358.05208	368.77942
400 KB	506.07064	487.70274	501.16744	487.63418	403.1689	411.93728
500 KB	556.69856	548.77324	545.0464	523.05374	437.0683	447.20134

Table 5. Comparison of different algorithms based on different factors

Factors	DES	3DES	AES	Blowfish
Evolution of Algorithms	In early 1970 by IBM and Published in 1977.	In 1978 developed by IBM	In 2001 developed by Vincent Rijmen, Joan Daeman	In 1993 developed by Bruce Schneier
Lengths of Key	64 (56 usable) Bits	112,168 Bits	128,192, 256 Bits	Variable key length i.e. 32–448 bits
Rounds in Algorithms	16	48	10,12,14	16
Block Size (Bits)	64	64	18	64
Security level	Sufficient Security	Sufficient Security	Excellent Security	Highly Secure
Encryption Speed	Slow	Very slow	Faster	Very fast

5. CONCLUSION

Clouds are virtual data storage devices. To be more specific, a cloud is a networking infrastructure that stores some or all of a company's resources and network capabilities. The cloud business is expanding at a quicker rate than anticipated. The same network and security stack may be utilized to offer connection, security, and visibility whether your workload is on campus or in the cloud.

The ability to access data from any platform makes it highly valuable; yet, because this type of facility requires internet connectivity, it is vulnerable to unwanted access.

For this type of vulnerability, data encryption is a common remedy. Two well-known methods for data encryption and decryption are the blowfish algorithm and AES-256. The research provided a possible approach to improve data security. The hybrid method encrypts data using AES-256. This encrypted data is sent via blowfish and then encrypted a second time.

This technique has the potential to be useful in situations when data is critical to an organization's success. One of the potential applications of this hybrid method is the storage of military data. his method improves data integrity by increasing encryption.

The suggested solution gives feather platform independence so that it may be utilized on a variety of devices. Future research will focus on optimal methods in terms of time and data compression.

REFERENCES

[1] "Cloud computing security", Wikipedia, Wikimedia Foundation, 4 March 2021, https://en.wikipedia.org/wiki/Cloud%20_computing%20security.

[2] A. Singh and K. Chatterjee, "Cloud security issues and challenges: A survey," *J. Netw. Comput. Appl.*, vol. 79, pp. 88–115, 2017.

[3] H. Wu, Y. Ding, C. Winer, and L. Yao, "Network security for virtual machine in cloud computing," in *5th International Conference on Computer Sciences and Convergence Information Technology*, 2010, pp. 18–21.

[4] A. Singh and K. Chatterjee, "Cloud security issues and challenges: A survey," *J. Netw. Comput. Appl.*, vol. 79, pp. 88–115, 2017, (ISSN: 1084-8045), http://doi.org/10.1016/j.jnca.2016.11.027.

[5] J. Wu, et al., "Cloud storage as the infrastructure of cloud computing," in *2010 International Conference on Intelligent Computing and Cognitive Informatics*, IEEE, 2010.

[6] A. R. Riddle and S. M. Chung, "A survey on the security of hypervisors in cloud computing," in *2015 IEEE 35th International Conference on Distributed Computing Systems Workshops*, IEEE, 2015.

[7] P. M. Mell and T. Grance, "The NIST definition of cloud computing (SP800-145)," Tech. Rep., National Institute of Standards & Technology, Gaithersburg, MD, United States, 2011, URL http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf [Accessed: Feb. 10, 2018].

[8] F. Liu, J. Tong, J. Mao, R. Bohn, J. Messina, L. Badger, and D. Leaf, *NIST Cloud Computing Reference Architecture (SP 500-292)*, National Institute of Standards & Technology, Gaithersburg, MD 20899-8930, USA, 2011, URL http://ws680.nist.gov/publication/get_pdf.cfm?pub_id=909505.

[9] R. Kumar and R. Goyal, "On cloud security requirements, threats, vulnerabilities and countermeasures: A survey," *Comput. Sci. Rev.*, vol. 33, pp. 1–48, 2019.

[10] P. R. Kumar, P. H. Raj, and P. Jelciana, "Exploring data security issues and solutions in cloud computing," *Proced. Comput. Sci.*, vol. 125, pp. 691–7, 2018.

[11] Y. Jararweh, et al., "High-throughput encryption for Cloud computing storage system," *Int. J. Cloud Appl. Comput. (IJCAC)*, vol. 4, no. 2, pp. 1–14, 2014.

[12] M. Rady, T. Abdelkader, and R. Ismail, "Integrity and confidentiality in cloud outsourced data," *Ain Shams Eng. J.*, vol. 10, no. 2, pp. 275–85, 2019.

[13] S. Singh and S. Thokchom, "Public integrity auditing for shared dynamic cloud data," *Proced. Comput. Sci.*, vol. 125, pp. 698–708, 2018.

[14] R. Seiger, S. Groß, and A. Schill, "SecCSIE: A secure cloud storage integrator for enterprises," in *2011 IEEE 13th Conference on Commerce and Enterprise Computing*, IEEE, 2011.

[15] G. Ramachandra, M. Iftikhar, and F. A. Khan, "A comprehensive survey on security in cloud computing," *Proced. Comput. Sci.*, vol. 110, pp. 465–72, 2017.

[16] S. Singh, Y.-S. Jeong, and J. H. Park, "A survey on cloud computing security: Issues, threats, and solutions," *J. Netw. Comput. Appl.*, vol. 75, pp. 200–22, 2016.

[17] M. Kazim and S. Y. Zhu, "A survey on top security threats in cloud computing," 2015.

[18] L. Coppolino, et al., "Cloud security: Emerging threats and current solutions," *Comput. Electr. Eng.*, vol. 59, pp. 126–40, 2017.

[19] A. O. Joseph, J. W. Kathrine, and R. Vijayan, "Cloud security mechanisms for data protection: A survey," *Int. J. Multimedia Ubiquitous Eng.*, vol. 9, no. 9, pp. 81–90, 2014.

[20] F. Shahzad, "State-of-the-art survey on cloud computing security challenges, approaches and solutions," *Proced. Comput. Sci.*, vol. 37, pp. 357–62, 2014.

[21] R. Shaikh and M. Sasikumar, "Data classification for achieving security in cloud computing," *Proced. Comput. Sci.*, vol. 45, no. C, pp. 493–8, 2015.

[22] R. Patel and S. Dehariya, *Secure Model for Cloud Computing by Using Data Classification Methodology*, 2016.

[23] J. Wu, Z. Zhu, and S. Guo, "A quality model for evaluating encryption-as-a-service," in *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*, Springer, Cham, 2017.

[24] V. Kumar, et al., "Fully homomorphic encryption scheme with probabilistic encryption based on Euler's theorem and application in cloud computing," *Big Data Analytics*, pp. 605–11, 2018, Springer, Singapore.

[25] C. A. Dhote, "Homomorphic encryption for security of cloud data," *Proced. Comput. Sci.*, vol. 79, pp. 175–81, 2016.

[26] Y. Xu, et al., "Verifiable public key encryption scheme with equality test in 5g networks," *IEEE Access*, vol. 5, pp. 12702–13, 2017.

[27] V. Chang, Y.-H. Kuo, and M. Ramachandran, "Cloud computing adoption framework: A security framework for business cloud s," *Future Generation Comput. Syst.*, vol. 57, pp. 24–41, 2016.

[28] S. Wang, et al., "An efficient file hierarchy attribute-based encryption scheme in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 6, pp. 1265–77, 2016.

[29] S. Cao, et al., "Access control of cloud user's credible behavior based on IPv6," in *2018 IEEE 3rd International Conference on Big Data Analysis (ICBDA)*, IEEE, 2018.

[30] Z. Yan, et al., "Flexible data access control based on trust and reputation in cloud computing," *IEEE Trans. Cloud Comput.*, vol. 5, no. 3, pp. 485–98, 2015.

[31] N. Eltayieb, et al., "An efficient attribute-based online/offline searchable encryption and its application in cloud -based reliable smart grid," *J. Syst. Archit.*, vol. 98, pp. 165–72, 2019.

[32] A. G. Palathingal, et al., "Enhanced cloud data security using combined encryption and steganography," *Int. Res. J. Eng. Technol. (IRJET)*, vol. 5, no. 03, 2018.

[33] H. Hammami, et al., "Using homomorphic encryption to compute privacy preserving data mining in a cloud computing environment," in *European, Mediterranean, and Middle Eastern Conference on Information Systems*, Cham, Springer, 2017.

[34] Z. Huang, et al., "Insight of the protection for data security under selective opening attacks," *Inf. Sci.*, vol. 412, pp. 223–41, 2017.

[35] T. Ramaporkalai, "Security algorithms in cloud computing," *Int. J. Comput. Sci. Trends Technol. (IJCST)*, vol. 5, no. 2, pp. 500–3, 2017.

[36] N. Kaaniche and M. Laurent, "Data security and privacy preservation in cloud storage environments based on cryptographic mechanisms," *Comput. Commun.*, vol. 111, pp. 120–41, 2017.

[37] M. Mohis and V. S. Devipriya, "An improved approach for enhancing public cloud data security through steganographic technique," in *2016 International Conference on Inventive Computation Technologies (ICICT)*, vol. 3, IEEE, 2016.

[38] I. Ahmed, "A brief review: security issues in cloud computing and their solutions," *Telkomnika*, vol. 17, no. 6, pp. 2812–7, 2019.

[39] D. Pei, X. Guo, and J. Zhang, "A video encryption service based on cloud computing," in *2017 7th IEEE International Conference on Electronics Information and Emergency Communication (ICEIEC)*, IEEE, 2017.

[40] K. R. Sajay, S. S. Babu, and Y. Vijayalakshmi, "Enhancing the security of cloud data using hybrid encryption algorithm," *J. Ambient Intell. Humaniz. Comput.*, pp. 1–10, 2019.

[41] M. Babenko, et al., "Security analysis of homomorphic encryption scheme for cloud computing: Known-plaintext attack," in *2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, IEEE, 2018.

[42] N. S. Darwazeh, R. S. Al-Qassas, and F. AlDosari, "A secure cloud computing model based on data classification," *Proced. Comput. Sci.*, vol. 52, pp. 1153–8, 2015.

[43] M. G. Singh, M. A. Singla, and M. K. S. Sandha, "Cryptography algorithm comparison for security enhancement in wireless intrusion detection system," *Int. J. Multidiscip. Res.*, vol. 1, no. 4, pp. 143–51, 2011.

[44] M. Dworkin, "Recommendation for block cipher modes of operation. methods and techniques," *National Inst of Standards and Technology Gaithersburg MD Computer security Div*, 2001.

[45] Y. Kumar, R. Munjal, and H. Sharma, "Comparison of symmetric and asymmetric cryptography with existing vulnerabilities and countermeasures," *Int. J. Comput. Sci. Manage. Stud.*, vol. 11, no. 03, pp. 60–3, 2011.

[46] A. D. S. Elminaam, H. M. A. Kader, and M. M. Hadhoud, "Performance evaluation of symmetric encryption algorithms," *Commun. IBIMA*, vol. 8, no. 8, pp. 58–64, 2009.