Indonesian Journal of Computer Science
ISSN 2549-7286 (*online*)
Jln. Khatib Sulaiman Dalam No. 1, Padang, Indonesia
Website: ijcs.stmikindonesia.ac.id | E-mail: ijcs@stmikindonesia.ac.id

An Encryption Approach Based on Formal Method for Securing Distributed Big Data Storage in Cloud Environment

Anah Hassan Bijik[1], Souley Boukari[2], Abdulsalam Yau Gital[3], Mohammed Abdulhamid[4]
annebjk@gmail.com, bsouley2001@yahoo.com , asgital@gmail.com , yidi76@yahoo.com
[1] Department of Mathematical Sciences, Abubakar Tafawa Balewa University, Bauchi State, Nigeria.

Article Information

Submitted: 5 Feb 2023
Reviewed: 8 Feb 2023
Accepted : 27 Feb 2023

Keywords

Cloud Computing, Encryption, cloud data security, Big data

Abstract

Cloud computing is the technology used to store massive volumes of data (Big data) on servers whose whereabouts are hidden from user of the service. The possibility of sensitive data being accessed by cloud operators is a big source of worry regarding security and privacy. This makes Cloud computing adoption by organizations with users' sensitive details including the banking sector and government agencies marred by resentment. Therefore, a cryptography approach is proposed, named Secure Efficient Distributed Storage(SecDcloud) model that is designed to obtain an efficient mass distributed service, and is supported by the Modified Alternative Data Distribution (MAD2) Algorithm Modified Secured Data distribution (MSED2) Algorithm as well as Improved Data Conflation (IDCon) Algorithm based on formal method. Our proposed mechanism aims to split, encrypt sensitive data and store the data to the different cloud servers without causing big overheads using formal method which prevents cloud service providers from directly accessing the user's data. Our experiments, evaluated on security and efficiency of our technique, and is compared with state-ofthe-art Advanced Encryption Standard (AES) Algorithm and the results show that it is capable of effectively defending against the most common cloud-based threats while still maintaining a reasonable amount of processing time.

A. Introduction

Cloud computing is a broad and diverse field. The main advantage of cloud computing is that it eliminates the need for users to be in the same location where hardware software and storage space are physically present [1]. The cloud makes it possible to store and access data from anywhere and anytime without worrying about the maintenance of hardware software and storage space. All of these services are provided to user at a low cost [2] [1]. The user must pay according to the storage space in use. Due to this flexibility everyone is transferring data to cloud, in addition users are allowed to store large amount of data on cloud storage for future use [3] [4]. Many cloud users concern about their sensitive data to which the cloud operators have the access [5] [6] [7].

Security is major concern to the cloud computing. There is strong thrust to provide security at infrastructure -network level, Host level, application level and data. The data is associated with each level like network, host and Application level. The security issues related to different type attacks related to several technologies needs to be addressed [8] [9] [1]. Some security issues in cloud computing includes Availability, Third-Party Control, Data remanence, Legal Issues and Privacy.

Major privacy concerns related to cloud computing are sighted by Pearson. The various security issues related to data security, privacy, confidentiality, integrity and authentication needs to be addressed. Most of the cloud service providers store the data in plaintext format and user need to use their own encryption algorithm to secure their data if required [8]. The data needs to be decrypted whenever it is to be processed [3].

Cryptography includes techniques such as microdots, merging words with images, and other ways to hide information in storage or transit. However, in today's computer-centric world, cryptography is most often associated with scrambling plain text into a process called encryption, then back again. Modern cryptography in computer science concerns itself with the various data security [10]. In General, when data is encrypted, it is not easily understood by unauthorized people and to get plain text back decryption is used. For any kind of computation, one needs to perform the decryption first. Encryption solves major issues. But the power of cloud can be exploited if user is able to carry out computation on encrypted data [11] [12]. In addition, using encryption-focused techniques, it is hoped to secure data on cloud servers. Many researchers propose different cryptographic solution for cloud data protection, some of which are Fully Homomorphic Encryption (FHE) and Attribute Based Encryption(ABE). Although, this type of secure mechanisms can protect data effectively from the target attackers, however, the performance of the data processing can be negatively impacted due to the additional computations necessary in large data [13] [5]. Some operations cannot be accomplished because of the technical obstacles, such as noises in FHE [14] [15]. Approaches for solving the security issues on cloud-side are categorized in to two, using regulatory compliance mechanisms to restrict employees' behaviours' [16] [17] and preventing data from information leakage by encryptions, such as FHE and ABE. But this type of data security cannot satisfy most current industrial demands due to the lowered operation efficiency level and unsolved problems inside the solutions.

Formal methods are mostly concerned with the application of mathematical technique to software design and implementation, formal methods are mathematically sound procedures created to aid in the creation of sophisticated computer-based systems; in theory, formal methods are intended to create flawless systems. Formal methods make use of logic, sets, sequence and maps and eliminates ambiguities, contradictions and incompleteness, in addition it supports abstraction and serves as a wonderful medium for modelling. The contributions of formal methods to security are enormous [18]. General-purpose formal techniques have been applied successfully to security issues; for example, model checking tools enabled to find unknown attacks in security protocols [19] for the Needham-Schroeder public-key authentication protocol [20]. Adopting formal methods could pose an advantage to data security challenge with regard to computational overhead. In order to prevent cloud operators from reading users' data with little overhead, this paper proposes a way employing formal methodology and cryptography to address the issue of cloud data storage security.

Statement of the Problem

The use of cloud computing opens up a variety of avenues for Web-based service offerings to cater to different needs. Yet, data security and privacy have grown to be a serious concern that limits the use of many cloud applications. [5]. Users are allowed to store large amount of data on cloud storage for future use. The various security issues related to data security, privacy, confidentiality, integrity and authentication need to be addressed. Most of the cloud service providers store the data in plaintext format and user need to use their own encryption algorithm to secure their data if required. Many researchers such as [6] proposed different methods of solving security issues in cloud, but the issues still require researchers' attention for efficient security measures considering latency and overhead

Aim and Objectives

The aim of this research is to develop cryptography approach based on formal method to secure distributed big data storage in cloud Environment such that cloud service operators will not reach users original data with minimum overhead and latency.

The specific objectives of this research are:

1. To develop an approach for protecting user data based on a formal method to secure distributed big data storage in cloud computing, coined Secured Efficient Distributed Cloud Storage (SecDcloud) a technique that can encrypt and store users' data in different cloud data centres such that cloud service operators cannot have access to the original users' data stored on Cloud.
2. To achieve high-level secured data storage, by designing a scheme to split data into cloud servers in which internal threat can neither abuse nor retrieve the information from the stored data in the event that service providers 'policies are not adhered to.

3. Evaluate Security Performance (Execution time and effect of data size) the proposed Secured Efficient Distributed Cloud Storage (SecDCloud), with Advance Encryption (AES).

B. Related Works

This section presents reviews of related literatures on cloud security issues. This holds up the demonstration of the proposed research background and theoretical foundation. Most layers of cloud computing, from networks to system managements, have been affected by security vulnerabilities [21] [22]. Due to the linkages between technical applications, many security challenges in networks and data storage also apply to cloud computing such as using Virtual Machine (VM). Numerous researchers looked at security issues and potential solutions from various angles. For instance, previous studies of data encryption focused on small-to-medium-size data, which does not work well for big data due to issues in the performance and scalability, this makes data security in cloud a concern. The on-demand nature of cloud computing faces various security threats such as data loss, data leakage, denial-of-Service(DoS), account or service traffic hijacking, and malicious insiders. A malevolent hacker may modify critical data due to a careless cloud service provider. To tackle such risk, the important data needs to be encrypted and the encryption keys must be protected [1] [23]. If an intruder gains access to a customer credentials stored on a cloud, it may eavesdrop on transactions and activities, maliciously manipulates the data, returns falsified information, and may redirect the customer to illicit sites. DoS attack is another major concern for cloud platforms because most of the organizations are dependent on 24/7 availability of one or more services [24]. Denial of one or more services may be costly to the customers especially, when they are billed based on disk-space consumption and computation cycles. Account or service hijacking is another major threat faced by cloud platforms. Hijacking a service allows a malicious person to sneak into crucial and sensitive areas of a deployed service which may lead to breaching the integrity, availability, and confidentiality of that service. A malicious insider, e.g., a current or former employee, a business partner, or a contractor, may gain access to the data, network or system for malicious purposes [23]. The situation gets worse when the cloud service provider is solely responsible for data security. Cloud platforms attract more attacks due to their distributed nature. It is desirable that the data (video contents in this context) is protected and may only be accessed in encrypted form. Various clouds such as Google App Engine and Amazon web services have experienced security threats during recent years. These security flaws are exploited by illegal users to steal either secret information or disturb the normal operation of internet.

A component of protecting data in the cloud is data management security, which frequently focuses on encryption setups or data classifications for security [25] [26]. Some approaches have been developed to ensure the secure query processing for Resource Description Framework (RDF), such as using eXtensible Access Control Markup Language (XACML) management policy [27]. Moreover, a selective data encryption is considered a way of reducing computing cost while protecting data in clouds. For example, classifying data in diverse ranks using searchable encryption is an approach for users to alter whether the data need to be

encrypted [28] [25]. The majority of current data management techniques, however, make the assumption that cloud operators will not misuse the data or will only have access to it. Even though the data are encrypted on the cloud side in some circumstances, information can still be retrieved.

Data storage monitoring and protection is another aspect of securing data in the cloud, where the actions of cloud operators are observed or investigated utilizing attribute-based encryption (ABE) as one of the methods to ensure the privacy of information when the data is shared among several clouds [24] [29]. However, the challenge is restricting cloud operators' access scale can also result in other problems, such as data integration and data intactness [30] [31]. In addition, the rate of operation failure will rise if cloud service providers (CSP) are blocked [32] [24].

Moreover, the information protections, such as access control mechanisms and trust management is considered. For instance, an approach was proposed to secure instant community data access using trust level classification methodologies [33]. Another recent research proposed an undercrossed access control scheme for securing multimedia big data in cloud computing, which used ontology-based authentication classifications [34].

Most elliptic curve or bilinear pairing-based authentication algorithms are created for clientserver environments. They are not feasible for direct application to distributed service environments, where multiple service providers compete with each other for provisioning of various services. The user must manage numerous secret keys that they have acquired from various service providers. All service providers must use the same secret key to fix this problem. However, if an adversary acquires the secret key, it may pose as a legitimate service provider to deceive the users. Moreover, an intruder who captures the secret key may acquire the session keys as well. After acquiring one or more session keys, the attacker may eavesdrop on sensitive information transmitted between the user and another service provider. Even though, these researches mainly focused on securing data transmissions and authentications. The approaches do not have much control when data are stored on the cloud-side. In addition, it is desired to protect data on cloud servers by using encryptionoriented approaches.

Previous researches have also addressed this field, such as Fully Homomorphic Encryption (FHE) [35] and ABE. Despite this type of secure mechanisms can effectively protect data from the target attackers, such as external malicious actions and internal improper operations; nonetheless, the efficiency of the data processing can be negative impacted due to the additional computations [36]. Some operations cannot even be accomplished because of the technical obstacles, such as noises in FHE [14].

Other Approaches to secure Mass distributed storage have been proposed, for instance, in [37], according to the sensitivity of sensitive data, a method is suggested that divides the file and stores the data on several cloud servers. The input file's classification as sensitive or non-sensitive is decided by the user. Different virtual machines (VMs) are utilized to store sensitive files, while just one VM is used for non-sensitive items. The files are encrypted before being uploaded to the cloud server using the Elliptic Curve Integrated Encryption Scheme (ECIES) method.

[38] proposed an algorithm that would generate keys on its own based on data input and then encrypt data using the keys generated. Data that has been encrypted will be uploaded to a cloud storage service, and the key will be safely stored on a local server for later decryption. [39], developed an intelligent security method known as intelligent framework for the security of healthcare data called as IFHDS. Column based technique is applied large scale data security in developed system that has lesser impact on data processing. The developed technique masks the personal data and encrypt the sensitive data. The sensitive data were split into various parts on the basis of sensitivity levels where every part was separately stored in the distributed Cloud storage. The developed method secured the sensitive data of patients but with a higher computational time. The data owners manage the file configuration of data level security.

[5] applied a distribution storage technique with effective security awareness to secure big data storage distribution in Cloud computing. The developed method made use of algorithms such as alternative data distribution, secure data distribution and efficient data conflation. The developed system divided the files and data were distributed separately in the distributed servers of Cloud. The alternative method was established to identify whether the data packets were split in order to lessen the operational time. The developed security aware method effectively defended the threats of the Clouds and reduced the computational time, distributes the file and saves the data independently in the blob and scattered cloud servers. The suggested approach, known as the Distributed Data & Storage (D2S) model, is primarily supported by our suggested algorithms, such as the Distributed & Store Algorithm (DS).

In summary, most current active approaches solving the data abuse on cloud-side have two alternatives. The first common approach is using regulatory compliance mechanisms to restrict employees' behaviours [17]. This type of paradigm is not well controlled by technical methods. The other method is preventing data from information leakage by encryptions, such as FHE and ABE. But this type of data security cannot satisfy most current industrial demands due to the lowered operation efficiency level and unsolved problems inside the solutions. Our proposed scheme is an attempt that is designed for big data-related applications in which required a higher-level security using formal method and cryptography. A formed distributed storage manner can enable the data to be secured in cloud severs.

C. Methodology

The paper is aimed at improving the security in distributed storage as much as possible. The following sequence of steps identifies the methodology adopted.

i. Problem formulation

ii. Study Existing Encryption Algorithm and Security Aware -Efficient Distributed Storage (SA-EDS) and Model for Encryption /Decryption based on the existing algorithm

iii. Implement a cryptographic approach (SecDcloud) Algorithm based on formal method (Sequence)

iv. Evaluate the security and performance and Security of the SecDcloud compared with AES and (SA-EDS)

1. Problem Description

Given the initial input data and storage cloud servers. The goal is to identify a solution that can successfully store on cloud servers and ensure that cloud operators cannot access the data without greatly increasing execution time. The inputs include the initial data that consist of a string of user data packets. The outputs are two separate data packets that will be transmitted to different cloud storage servers. The newly created data packets must conceal user information so that cloud operators, who have access to the data, cannot read and comprehend it.

2. Proposed Secured Efficient Distributed Data Storage (SecDCloud) Scheme

This research focused cloud storage security issues and proposes a cryptography approach based on Formal Methods, by which the cloud service operators cannot directly reach partial data. The proposed techniques will perform sequence operation (head and last) to change the original data, divides the file, and store the split data in to different cloud servers in a distributed cloud. The proposed method is called A secured efficient distributed cloud storage (SecDCloud) model, which will be mainly supported by the proposed algorithms, including Modified Alternative Data Distribution (MAD2) algorithm, and Modified Secure Efficient Data Distributions (MSED2) Algorithm and Improve Efficient Data Conflation (IDCon) Algorithm, Figure 1, is the abstract model of the proposed scheme.

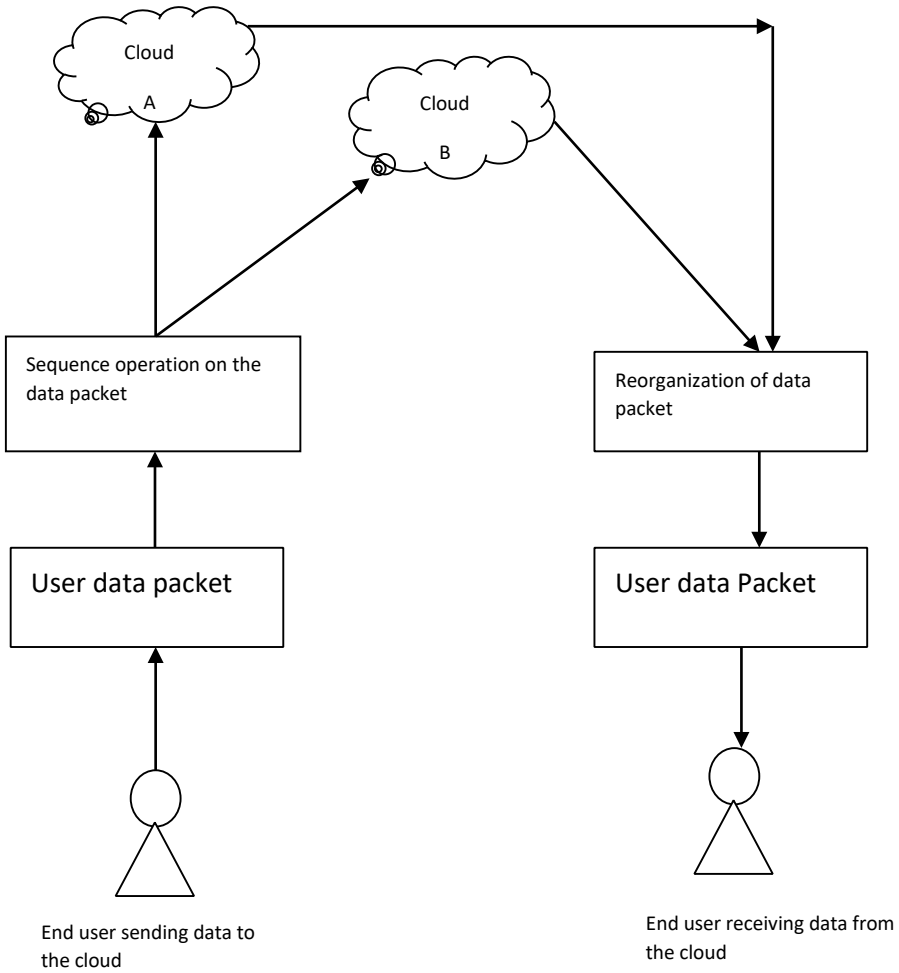


Figure 1. Proposed SecDCloud Abstract Model

Model Description

User data is partitioned in to four functional units and arrange in a sequence, and subject to sequence operation (Head, Last). The Head and the last are merge and encrypted as well middle. The encrypted data are then stored in different cloud. This method is proposed to avoid cloud data centres operators from having access to original user data. The main problem to be addressed here is security of user data in distributed storage.

The inputs include the initial data that consist of user data packets represented as terms in sequence. Sequence and its operations are a part of Model based Approach in adapting formal method which is the basis for this work. The outputs are two separate data packets that will be transmitted to different cloud storage servers. The new generated data packets need to hide user data so that the cloud operators cannot read and understand the information, even though they have the access to the data.

Model Description

User data is partitioned in to four functional units and arrange in a sequence, and subject to sequence operation (Head, Last). The Head and the last are merge and encrypted as well middle. The encrypted data are then stored in different cloud. This method is proposed to avoid cloud data centres operators from having access to original user data. The main problem to be addressed here is security of user data in distributed storage.

The inputs include the initial data that consist of user data packets represented as terms in sequence. Sequence and its operations are a part of Model based Approach in adapting formal method which is the basis for this work. The outputs are two separate data packets that will be transmitted to different cloud storage servers. The new generated data packets need to hide user data so that the cloud operators cannot read and understand the information, even though they have the access to the data.

Model Expression

This can be expressed in two folds Storage and Retrieval

Storage: Consider the Sequence

D = {d1, d2, d3, d4}

Where D is d1, d2, d3, d4 form the terms represent data packets which is in form of plain text Perform sequence operations [head, last] on D as well as middle

Such that

D1 = d1d4

D2 = d2d3

//When operations are executed and encrypted on the data packets, it generates

Then D1 and D2 are encrypted and sent to Cloud A and Cloud B

Retrieval (Merging and Reorganising)

Reorganising is performed with mid first, and Decrypted back to its original form

Operations: Mid first

Decrypt back to original form.

General Concept Pseudocode for SecDcloud

1. For all the terms d1, d2, d3, d4 in sequence D, as sensitive text (Assume that the plaintext is sensitive)
2. Perform operation [Head, Last] as well as middle such that d1=d1d4, D2=d2d3
3. Generate Key, XOR with D1, D2
4. 4. Store on Cloud A and B
5. End.

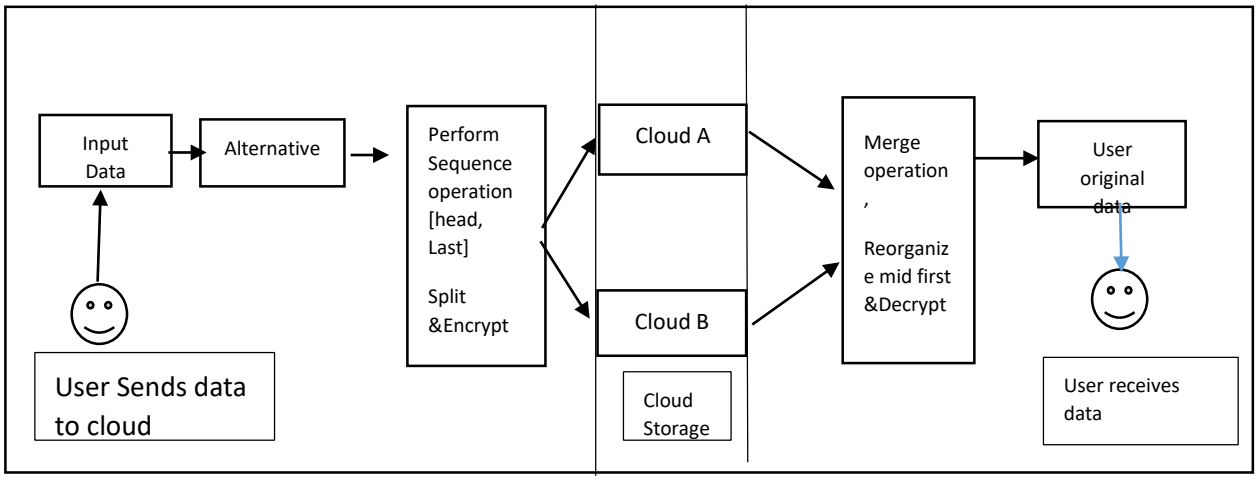


Figure 2. SecDcloud Workflow Structure

There are basically three phases in the model as shown in Figure 2, these phases form the crucial steps during the data transmissions. One is processing data into two separate data strings, this process is accomplished by pairing (Head, Last) of the sequence of data packets and encrypting, finally, both encrypted data packets were sent to separate cloud servers. Then the storage unto different cloud servers at different locations, and finally merging and reorganizing and decrypting the data to obtain the original data which forms the retrieval case. Attaining the original data needs a reorganization operation after the data packets are received from cloud sides. Then the key is XORed with the retrieved Mid first Operation Next, the original data will be gained after this procedure is finished.

Pseudo code and Algorithms

The main purpose for this algorithm is to store cloud data in a distributed manner such that cloud operators will not have access to client data in a cloud environment.

Modified Alternative Data Distribution Pseudo code (MAD2)

1. Input the searchable named-data-packets that are searchable and pre-named list (PNL).
2. For all named-data-packets, we search each data packet and see whether there is a name label that matches searchable labels in PNL.
3. If a match is found, execute Modified Efficient Data Distribution (MSED2) Algorithm by which the data packets are split into two parts and stored in distributed cloud servers. In this process, the split data packets include α and β.
4. Otherwise, execute an XOR operation to the data packet and generate an encrypted data packet D_{xor}.

5. Output the encrypted data packets, including D_{xor} , α, and β.

Modified Alternative Data Distribution (MAD2) Algorithm.
Require: NDP, PNL
Input NDP, PNL

1: For ∀ NDP do
2: for each data packet do
3: if ∃ a li ∈ PNL then
Execute MSED2 Algorithm / Algorithm */*
4:
5: else
6: Do XOR operation to the data packet
*7: /*Do XOR operation before the data packet is sent out*/*
8: Generate D xor
9: end if
10: end for
11: Obtain the values of D

Modified Efficient Data Distribution Algorithm (MSED2)

1. Input data D create an initialise a few data set α,β and assign 0 to them
2. Randomly generate a key *K* that is stored at the user's special register for the purpose of encryption and decryption. This is the crucial part for protecting privacy before the data are sent out.
3. Apply Sequence operation (Head, Last) also known as pairing which produces D1, D2
4. Execute the XOR operation to obtain α,β α □ D1□K β □ D2□K
   where k is the key randomly generated and stored
5. Output α and β and separately store them in the different cloud servers.

Modified Efficient Data Distribution (MSED2) Algorithm
Require: *D //non empty*

Ensure: α, β
1: Input D
2: Initialize $^{\alpha}\leftarrow 0, ^{\beta}\leftarrow 0$
3: Randomly generate a key K
*5: **for** all input data packets **do***
6: $^{\alpha}\leftarrow d1dn \square K$ */ * head, last*
7: $\beta \leftarrow d2dn\text{-}1 \square K$
*8: end **for***
9: output α β

Improved Data Conflation (IDCon) algorithm

Improved Data Conflation (IDCon) Algorithm is designed for users obtain their original data from distributed cloud servers.

Improved Data Conflation Pseudo codes

1. Input the data, α and β, that are acquired from different cloud servers. The user obtains the key *K* from the special register.
2. Initialize a few dataset γ, γ', and *D* for the operation needs.
3. Execute the XOR operation to both α and β by using *K*. Assign the value to γ and γ', respectively.
4. γ and γ and assign to *D*
5. output D

Improved Data Conflation (IDCon) Algorithm.

Require: α, β, K

Ensure: *D*

1: Input α, β, K

2: Initialize $\gamma \leftarrow 0, \gamma' \leftarrow 0, D \leftarrow 0$

*3: /*receives α, β from separate cloud servers mid first*/*

4: □□*α*□*k* , □'□*β*□*k*

5: *D* □ □□'

6: Output D

Implementation

The experimental environment was configured as follows. The Proposed Scheme was implemented and tested on a windows machine with a 9th gen corei7 processor, 16GB of RAM, 2Terabyte storage and windows 10 Operating system. Google Server at different locations was used Python was used as language of choice.

Experimental Settings

The proposed model was evaluated on different inputs size while assessing the execution time. The settings are as follows:

i. Setting 1: evaluations based on the data required to be encrypted. The assessment was processed by different input data sizes, as follows: Setting 1: 1 KB, Setting 2: 1 MB, Setting 3: 10 MB, Setting 4: 50 MB, Setting 5: 250 MB, Setting 6: 500 MB. ii. Setting 2: evaluations based on the data retrieved from cloud servers. The assessment was processed by different retrieval data sizes, as follows: Setting 1: 1 KB, setting 2: 1 MB, setting 3: 10 MB, Setting 4: 50 MB, Setting 5: 250 MB, Setting 6: 500 MB

In order to evaluate the expected performance dimensions, we evaluated the proposed model by assessing its execution time while different input data sizes were operated and compared with MSED2 and Advanced Encryption Standard (AES) Which is one of symmetric encryption that most used often and utilized popularly over the world as the most secure algorithm for encryption available today.

The experimental results are as follows:

Table 1. Encryption Result of MSED2 and AES at different settings

S/No	Settings (Data Size)	Secured Distributed Storage (SecDcloud) MSED2	Advanced Encryption Standard (AES) AES Encrypt
1	Setting 1-1	90	218
2	Setting 1-2	800	1600
3	Setting 1-3	5020	5210
4	Setting 1-4	26300	37000
5	Setting 1-5	147000	180000
6	Setting 1-6	283000	357000

The table 1 above represents the results obtained for Modified Secured Data Distributed (MSED2) Algorithm when compared with the Advance Encryption Standard (AES) which is widely accepted approach the result shows that our approach has less execution time at different settings.
The Results of decryption at different settings is presented in the table 2 below, in which comparison was drawn from our Data Conflation approach and AES decrypt. Generally, from the results we can deduce that the decryption takes longer when compared to encryption. For instance, at the initial settings, there is a difference of 800ms when compared with AES decrypt. As the data size increase at the last settings, we discover 12,000ms as the difference, which implies our approach even though longer performs better.

Table 1. E Decryption Result of IDCon and AES Decrypt at different settings

S/No	Settings (File Size)	Secured Efficient Distributed Storage (SecDcloud) IDcon -Decrypt	Advanced Encryption Standard (AES) AES Decrypt
1	Setting 2-1	103	1000
2	Setting 2-2	1000	1700
3	Setting 2-3	9000	10000
4	Setting 2-4	30000	40000
5	Setting 2-5	170000	240000
6	Setting 2-6	360000	480000

The few experimental results for performance Evaluations are represented with graphs, and observations discussed. The encryption execution time are

examined. For instance, Figures 3,4 &5 show the comparison of encryption execution time between MSED2 and AES using same sized input data generated for setting 1-1 to 1-6 According to the lines of graphs, the proposed scheme has shorter execution time when compared with AES which is the standard approach, longer decryption execution time was observed under the same settings.

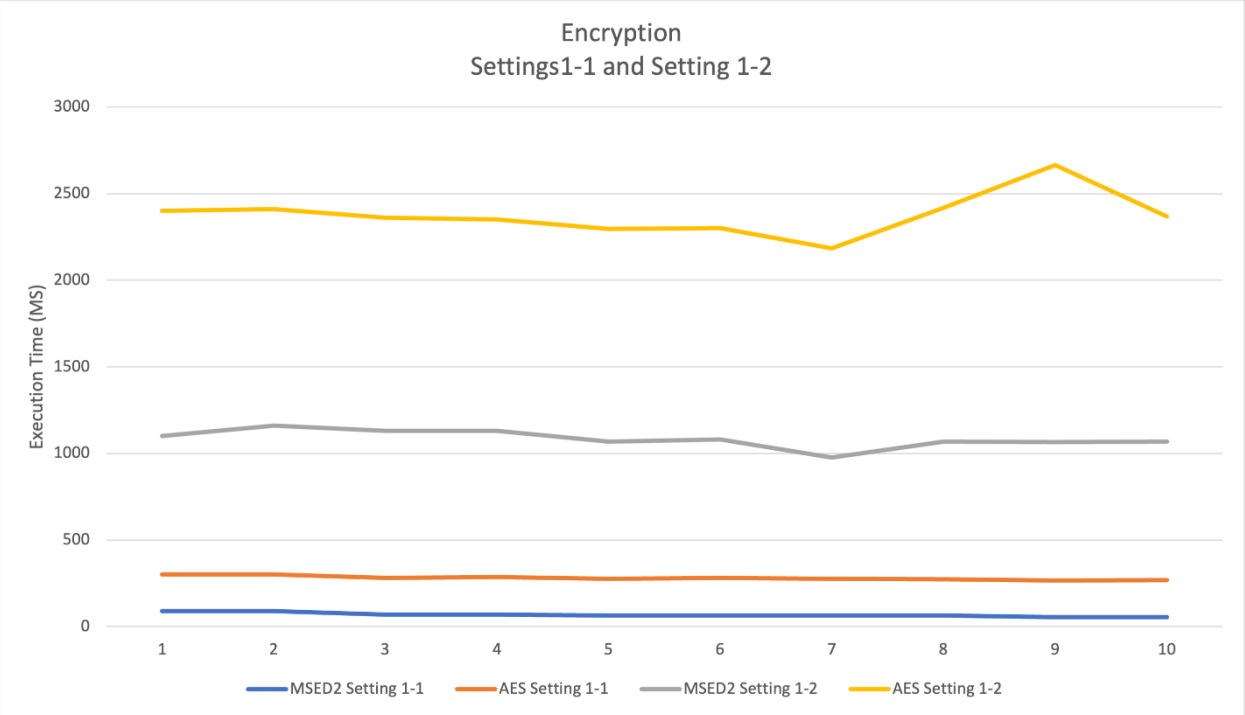


Figure 3. Comparison on the Execution (Encryption) time between MSED2 and AES (1-1,1-2)

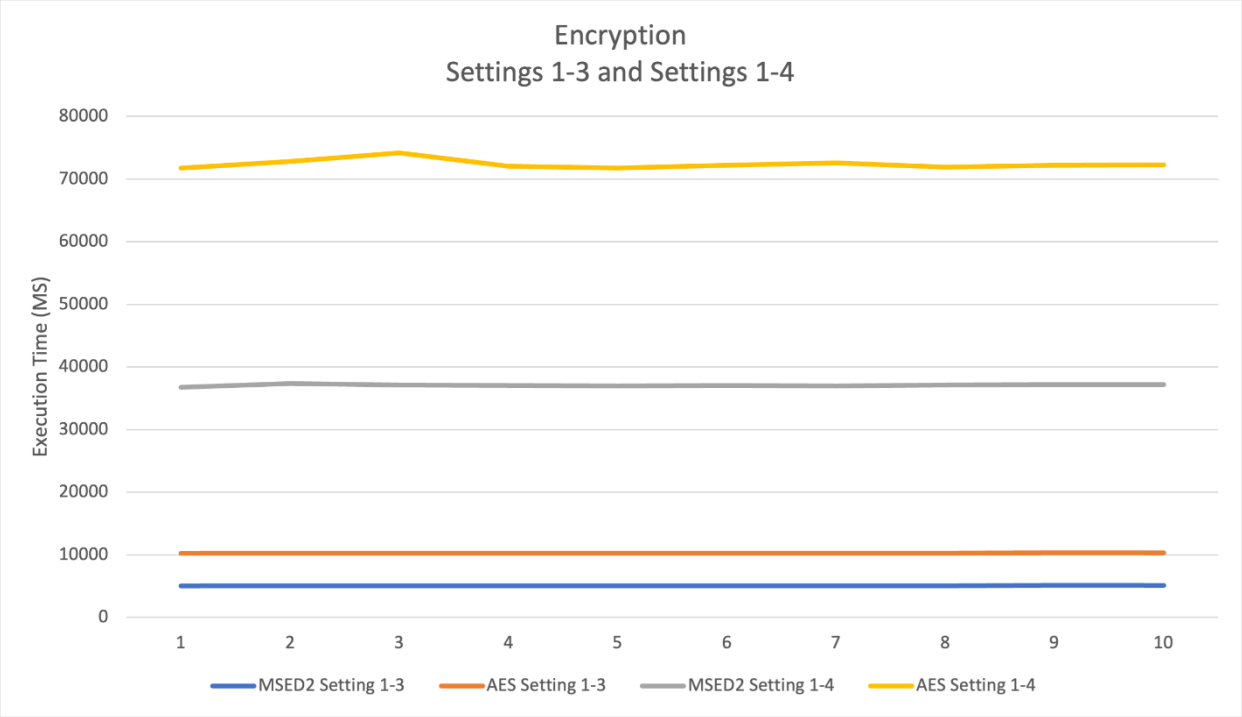


Figure 4. Comparison on the Execution (Encryption) time between MSED2 and AES (1-3, 1-4)

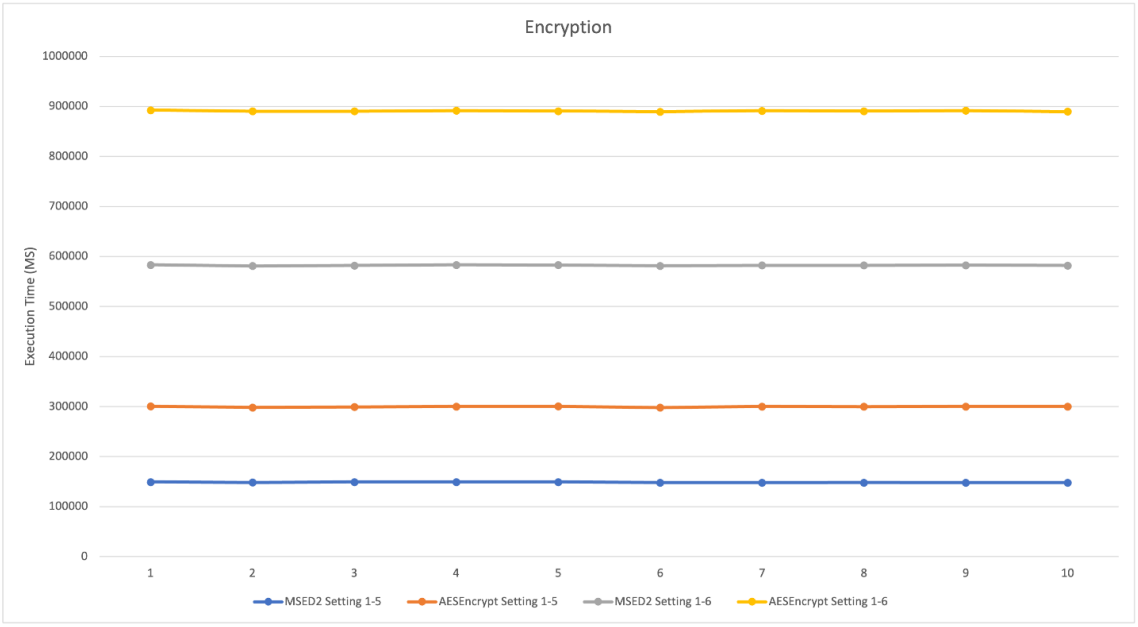


Figure 5. Comparison on the Execution (Encryption) time between MSED2 and AES (1-5, 1-6)

In figure 6 &7 the decryption time between IDCon and AES Decrypt, was considered

Represented execution time differences between the encryption and decryption when the data sizes were varied. The horizontal axis represents the number of the evaluations. The figure showed that the data that needed decryptions were impacted by the data size. The execution time became longer when the data size increased.

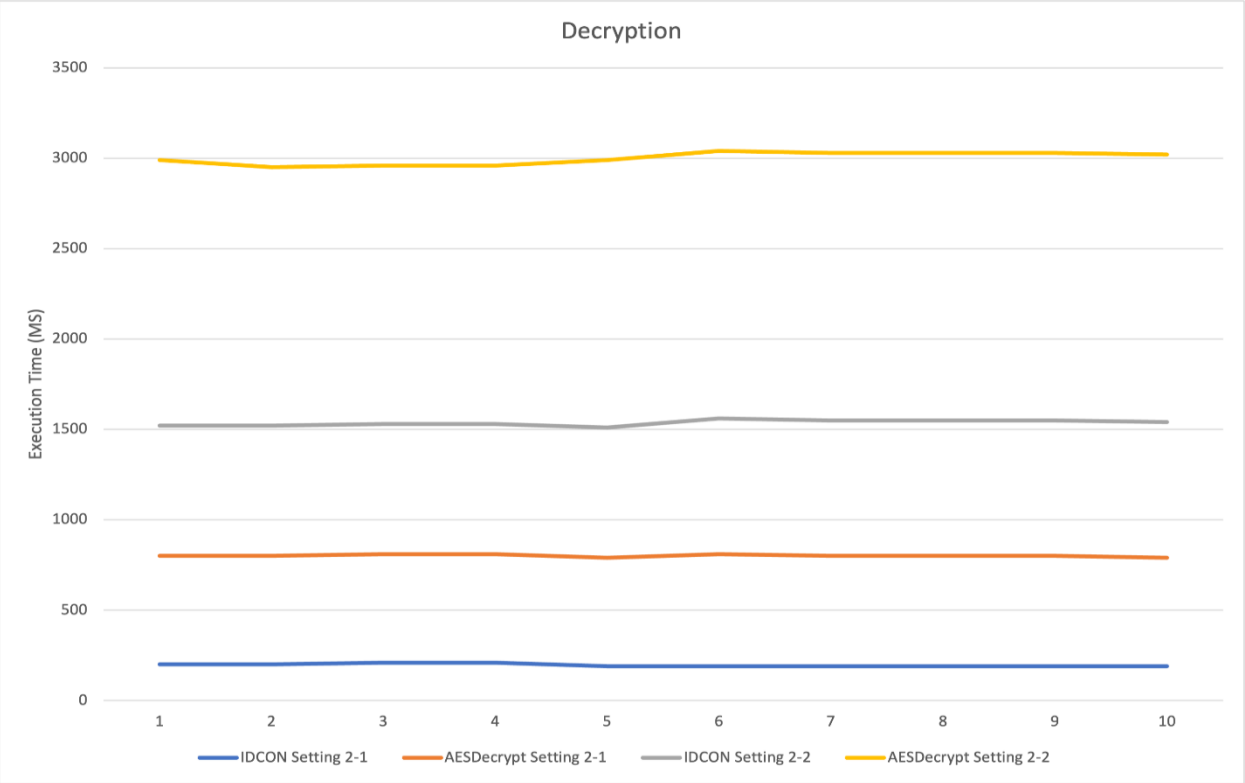


Figure 6. Comparison on the Execution (Decryption) Time Between MSED2 and AES (2-1, 2-2)

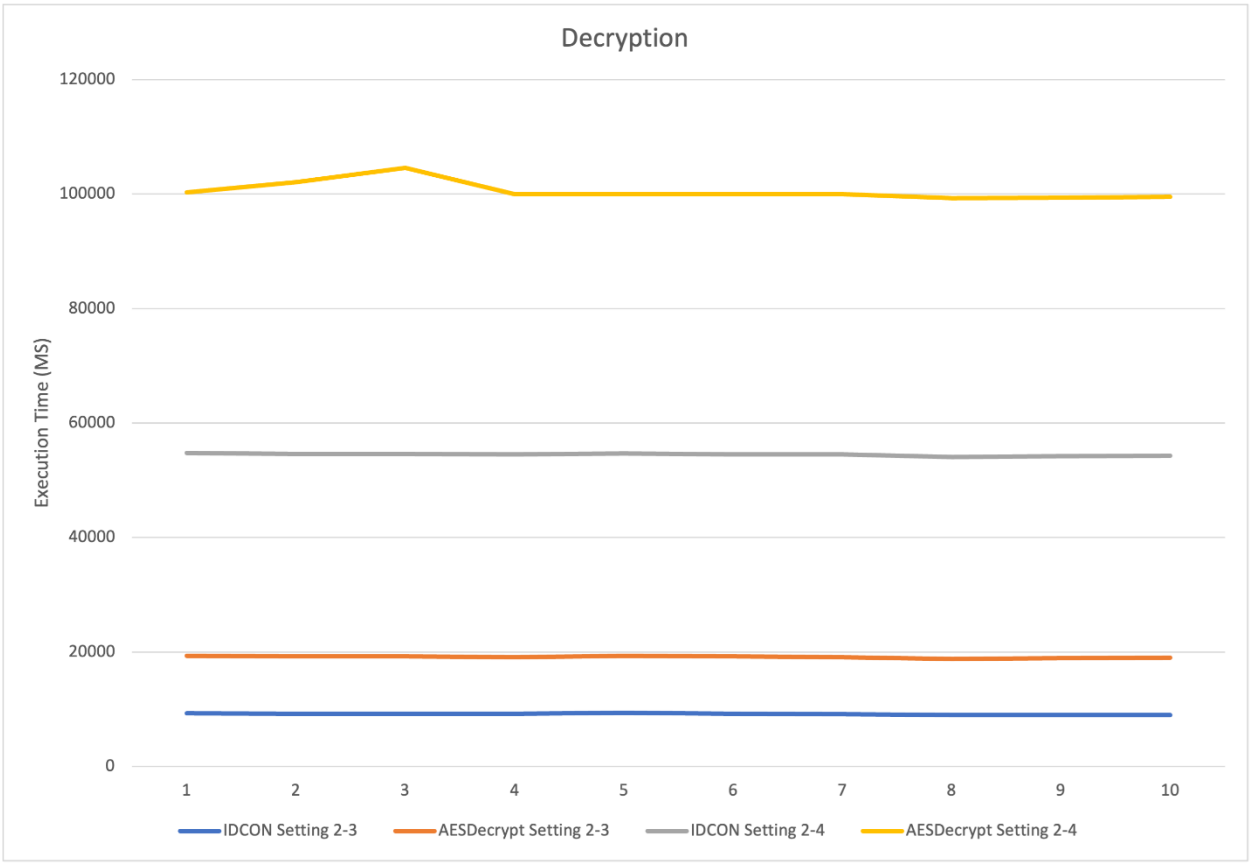
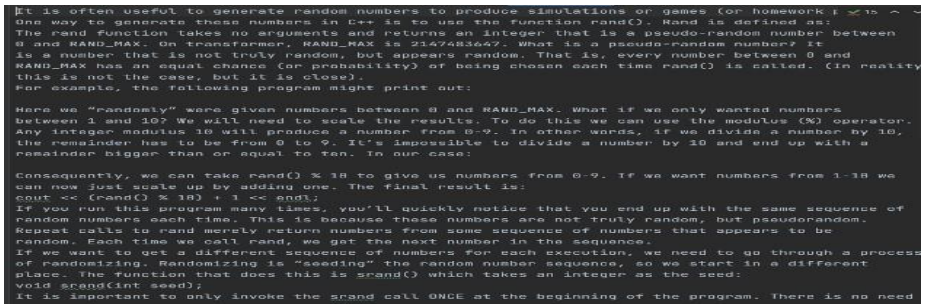


Figure 7. Comparison on the Execution (Decryption) time between MSED2 and AES (2-3, 2-4)

The File Content is encrypted using Visual Assessment as shown in Figure 8&9 shows an example of original file before encryption which is readable, while Figure 9. shows that content is well encrypted. This takes care of the threat from the cloud operators who may decide not to comply with the regulations provided by the service providers. This also implies that the scheme can protect data on transmissions, since the encryption will be performed before the split and sent to cloud.


It is often useful to generate random numbers to produce simulations or games (or homework
One way to generate these numbers in C++ is to use the function rand(). Rand is defined as:
The rand function takes no arguments and returns an integer that is a pseudo-random number between
0 and RAND_MAX. On transformer, RAND_MAX is 2147483647. What is a pseudo-random number? It
is a number that is not truly random, but appears random. That is, every number between 0 and
RAND_MAX has an equal chance (or probability) of being chosen each time rand() is called. (In reality
this is not the case, but it is close).
For example, the following program might print out:

Here we "randomly" were given numbers between 0 and RAND_MAX. What if we only wanted numbers
between 1 and 10? We will need to scale the results. To do this we can use the modulus (%) operator.
Any integer modulus 10 will produce a number from 0-9. In other words, if we divide a number by 10,
the remainder has to be from 0 to 9. It's impossible to divide a number by 10 and end up with a
remainder bigger than or equal to ten. In our case:

Consequently, we can take rand() % 10 to give us numbers from 0-9. If we want numbers from 1-10 we
can now just scale up by adding one. The final result is:
cout << (rand() % 10) + 1 << endl;
If you run this program many times, you'll quickly notice that you end up with the same sequence of
random numbers each time. This is because these numbers are not truly random, but pseudorandom.
Repeat calls to rand merely return numbers from some sequence of numbers that appears to be
random. Each time we call rand, we get the next number in the sequence.
If we want to get a different sequence of numbers for each execution, we need to go through a process
of randomizing. Randomizing is "seeding" the random number sequence, so we start in a different
place. The function that does this is srand() which takes an integer as the seed:
void srand(int seed);
It is important to only invoke the srand call ONCE at the beginning of the program. There is no need

Figure 8. Sample File Content Before Encryption

Figure 9. Sample File Content After Encryption

D. Conclusion

The paper is expected to address the problem of cloud data storage and is intended to offer a method that could prevent cloud operators from viewing sensitive user data. Addressing this goal, we proposed a novel cryptographic approach entitled as Secure Data Distributed Storage (SecDcloud) model. The model is supported by Modified Alternative Data
Distribution (MAD2), Modified Secure Data Distributions (MSED2) and Improved Data Conflation (IDCon) algorithms. Our experimental evaluations had proved that our proposed scheme could effectively defend major threats and from the results obtained will secure users data from both external and internal threats. In addition, the scheme performs better in terms of computing time compared to the Advanced Encryption Standard(AES).

E. References

[1] GITAL A. Y., ISMAIL A. S., CHEM M. A. & CHIROMA H., "Framework for the Design of Cloud Based Collaborative Virtual Environment Architecture," Proceedings of the International MultiConference of Engineers and computer Scientist, 2014.

[2] SONG D., SHI E., FISCHER I & SHANKAR U., "Cloud data protection for the masses," 2012.

[3] PANSOTRA E. A. & SINGH E. S. P., "Cloud Security Algorithms. International Journal of Security and its Applications," vol. 9, pp. 353-360, 2015.

[4] SURESH K. & PRASAD K., "Security Issues and Security Algorithms in Cloud Computing International Journal of Advanced Research in Computer Science and Software Engineering, 2.," 2012.

[5] LI Y., GAI K., QUI L., QIU M. & ZHAO H., "Intelligent cryptography approach for secure distributed big data storage in cloud computing," Information Sciences, vol. 387, pp. 103 -115, 2017.

[6] POTEY M. M., DHOTE C. & SHARMA D. H., "Hommorphic Encryption for security of Cloud Data. Procedia Computer Science.," vol. 79, p. 175 – 181, 2016.

[7] TEBAA M., EL HAJJI S. & EL GHAZI, "A Homomorphic encryption applied to the could computing security. Proceedings of the World Congress on Engineering," pp. 4 - 6, 2012.

[8] ATAYERO A. A. & FEYISETAN O., "Security Issues in cloud computing: The Potentials of homomorphic encryption," International Journal of Emerging Trends in Computing and Information Sciences, vol. 2, pp. 546 - 552, 2011.
[9] WANG B., LI M., CHOW S. S. & LI H., "Computing encrypted cloud data efficiently under multiple keys. Communications and Network Security (CNS)," IEEE Conference, pp. 504 - 513, 2013a.
[10] LIN H. Y., & TZENG W. G., "A secure erasure code-based cloud storage system with secure data forwarding," IEEE transactions on parrell and distributed systems, vol. 23, pp. 995 - 1003, 2012.
[11] WEI L., ZHU H., CAO Z., DONG X., JIA W., CHEN Y. & VASILAKOS A. V., "Security and privacy for storage and computation in cloud computing, Information science," Information science, vol. 258, pp. 371 - 386, 2014.
[12] WANG C., WANG Q., REN K. & LOU. W., "Privacy-preserving public auditing for data storage security in cloud computing.," INFOCOM,2010 Proceedings IEEE, pp. 1 - 9., 2010.
[13] VAN DIJK M. & JUELS A., "On the Impossibility of Cryptography Alone for Privacy Preserving Cloud Computing," September 2010. [Online]. Available: https://www.researchgate.net/publication/220335789_On_the_Impossibili ty_of_Cryptography_Alone_For_Privacy_Preserving_Cloud_Computing. [Accessed 15 December 2022].
[14] YAGISAWA M., "Fully Homomorphic Encryption without bootstrapping," IACR Cryptology eprint Archive, p. 474, 2015.
[15] BRAKERSKI X., "Fully homomorphic encryption without modulus switching from classical GapSVP," Advances in Cryptology-CRYPTO, 2012.
[16] HERRERA-VIEDMA E., CABRERIZO F. J., KACPRZYK J., & PEDRYC W., "A review of soft consensus models in a fuzzy environment.," Information Fusion, vol. 17, pp. 4 - 13, 2014.
[17] MODI C., PATEL D. C., BORISANIYA B. O., PATELS A. & RAJARAJAN M., "A survey on security issues and Solutions at different layers of Cloud computing.," Journal of Supercomputing, vol. 63, pp. 561 - 592, 2013.
[18] WING J. M. "A symbiotic Relationship Between Formal Methods and Security pg. 2638, 1998," in In Proceedings of the ONR/SNF Workshop on Computer Security, Dependability, and Assurance: From Needs to Solution. Also available as Carnegie Mellon University report CMU-CS-98-188, Washington DC, USA,, 1998.
[19] GAVIN L., "Breaking and Fixing the Needham – Schroeder public-key Protocol Using FDR," vol. 1066, pp. 147 - 166, 1996.
[20] NEEDHAM R. M. & SCHROEDER M. D., "Authentication Revisited, Operating System Review," p. 21(1), 1987.
[21] PEDRAYCZ W., "Allocation of information granularity in optimization and decisionmaking models; towards building the foundations of granular computing," European Journal of Operational Research, vol. 232, pp. 137 - 145, 2014.
[22] WANG C., CHOW S. S., WANG Q., REN K., & LOU W., "Privacy-preserving public auditing for secure cloud storage. IEEE Transactions on Computers," vol. 62, pp. 362 - 375, 2013b.
[23] ALAHMADI A., ABDELHAKIM M., REN J. & LI T., "Defense against primary user emulation attacks in cognitive radio networks using advanced encryption

standard.," IEEE Transactions on Information forensics and security, vol. 9, pp. 772 - 781, 2004.
[24] GAI K., QUI M., THURAISINGHAM B. & TAO L., "Proactive attribute-based secure data schema for Mobile cloud in financial industry. High performance Computing and Communications (HPCC)," IEEE 7TH International Symposium on Cyberspace Safety and Security (CESS), pp. 1332 - 1337, 2015b.
[25] LIU Q., WANG G. & WU. J., "Time-based proxy re-encryption scheme for secure data sharing in a cloud environment: Information Sciences," vol. 258, pp. 355 - 370, 2014.
[26] DAYANDANDA R. & SOMESWAR G. M., "Time-Based Proxy Re-encryption Scheme for Secure Data Sharing in a Cloud Environment," International Journal of Emerging trends in Science and Technology,2, 2015.
[27] CHADWICK D.W. & FATEMA K., "A primary preserving authorization system for the cloud.," Journal of computer and system sciences, vol. 78, pp. 1359 - 1373, 2012.
[28] CAO N., WANG C., LI M., REN K. & LOU W., "Privacy- preserving multi-keyword ranked search over encrypted cloud data," IEEE Transactions on Parallel and distributed systems, Vols. 222 - 233, p. 25, 2014.
[29] LI M. S., ZHENG Y., REN K. & LOU W., "Scalable and secure sharing of personal health
records in Cloud computing using attribute-based encryption," IEEE transactions on parallel and distributed, vol. 24, pp. 131 -143, 2013.
[30] AGRAWAL R. & JOHNSON C., "Securing Electronic Health Records without impeding the flow of Information. International Journal of Medical informatics," International Journal of Medical Informatics, vol. 76, pp. 471 - 479, 2007.
[31] GAI K., MING Z., ZHAO H. & QUI M., "Intercrossed Access Controls for Secure Financial Services on Multimedia Big. Data in Cloud System ACM Transactions on Multimedia Computer, communications, and Applications (TOMM)," vol. 12, p. 67, 2016a.
[32] CHEN C. P., & ZHANG C. Y., "Data-intensive applications, challenges, techniques and Technologies A survey on Big Data.," Information Sciences, vol. 181, pp. 935 - 953, 2014.
[33] YAN Z., WANG M., & ZHANG P. A., "Scheme to secure Instant Community Data Access Based on trust and Contexts, Computer and information Technology (CIT), 2014 IEEE International Conference on," IEEE, pp. 646 - 651, 2014a.
[34] LI Y., GAI. K., QUI M. & ZHAO H., "Intelligent Cryptography approach for secure distributed big data storage in cloud computing.," Information Sciences, 2016. [35] PLANTARD T., SUSILO W. & ZHANG Z., "Fully homomorphic encryption using hidden ideal lattice," IEEE transactions on information forensics and security, vol. 8, pp. 2127 - 2137, 2013.
[36] ALIEV R., PEDRYCZ W., FAZLOLLAHI B., HUSEYNOV O. H., ALIZADEH A.V., & GUIRIMOV B., "Fuzzy Logic-based generalized decision theory with imperfect information," Information Sciences, pp. 18 - 42, 2012.
[37] GOKULA KRISHNAN V., DEEPA J., VENKATA L. S., MOHANA PRAKASH T. A., SREERAMA MURTHY K. & DIVYA V., "Securing Mass Distributed Big Data Storage using Intelligent Elliptic Curve Integrated Encryption Scheme in Multi-Cloud

Computing," International Journal of Engineering Trends and Technology, vol. 70, no. 1, pp. 35 - 42, 2022.
[38] TAJAMMUL M. & PARVEEN R. "Auto encryption algorithm for uploading data on cloud storage," International Journal of information Technology, vol. 12, pp. 831 - 837, 2020.
[39] ESSA Y.M., HEMDAN EL-DIN E., ELMAHALAWY A., ATTIYA G., & EL-SAYED A., "IFHDS: Intelligent framework for securing healthcare bigdata. Journal of medical systems," Journal of Medical System, vol. 43, pp. 1-13, 2019.