Computer Systems
Science and Engineering

Tech Science Press

DOI: 10.32604/csse.2023.029790
Article

Improved Cloud Storage Encryption Using Block Cipher-Based DNA Anti-Codify Model

E. Srimathi[1],* and S. P. Chokkalingam[2]

[1]SRM Institute of Science and Technology, Ramapuram Campus, Chennai, 600089, Tamilnadu, India
[2]Department of CSE, Saveetha Institute of Medical and Technical Sciences, 602105, Tamilnadu, India
*Corresponding Author: E. Srimathi. Email: srimathikarthick1@gmail.com

Received: 11 March 2022; Accepted: 08 June 2022; Published: 26 May 2023

Abstract: When it comes to data storage, cloud computing and cloud storage providers play a critical role. The cloud data can be accessed from any location with an internet connection. Additionally, the risk of losing privacy when data is stored in a cloud environment is also increased. A variety of security techniques are employed in the cloud to enhance security. In this paper, we aim at maintaining the privacy of stored data in cloud environment by implementing block-based modelling to boost the privacy level with Anti-Codify Technique (ACoT) and block cipher-based algorithms. Initially, the cipher text is generated using Deoxyribo Nucleic Acid (DNA) model. Block-cipher-based encryption is used by ACoT, but the original encrypted file and its extension are broken up into separate blocks. When the original file is broken up into two separate blocks, it raises the security level and makes it more difficult for outsiders to cloud data access. ACoT improves the security and privacy of cloud storage data. Finally, the fuzzy-based classification is used that stores various access types in servers. The simulation results shows that the ACoT-DNA method achieves higher entropy against various block size with reduced computational cost than existing methods.

Keywords: Cloud storage; cloud computing; block cipher; anti-codify technique

1 Introduction

The cloud model, which makes use of a shared resources pool (servers, networks, storage, etc.), enables ubiquitous on-demand network connectivity that can be accessed from any location at any time [1]. Also included is a new IT infrastructure, which allows for modification and updating, increased resource utilisation, reduced computing, and a storage platform for the purpose of using information technology in the business sector. Many significant disadvantages to the cloud model exist, including the requirement for data centres to keep sensitive outsourced data in a secure manner and to ensure the data integrity [2–4]. At various architectural layers (platform, infrastructure, and application), it is possible for the cloud to suffer from a range of vulnerabilities as a result of design, programming, or configuration errors made by service providers or designers [5].

To make matters worse, terrorists have discovered that the cloud is a popular target for their nefarious activities. Cloud security must be adequately implemented in order to make accurate predictions about the size of the cloud market. It is concluded that cloud computing security is the most important concern [6]. The use of commercial public cloud storage for particularly sensitive information may necessitate the requirement for data owners to encrypt their data in order to prevent it from being made available to unauthorised third parties [7].

By virtue of this development, traditional plaintext keyword search-based data utilisation services will be rendered obsolete, as would be expected [8]. The massive expenses associated with data transmission capacity in cloud-scale frameworks make downloading and decrypting all data locally an increasingly impractical proposition as time goes on [9]. Finding a reliable search engine and safeguarding encrypted cloud data are both crucial in this regard. For modern distributed computing systems, implementing an encrypted cloud data search framework is still a tough challenge due to the existence of several unbreakable security and protection constraints such as data protection, index privacy and keyword privacy constraints.

In this paper, we implement and use block-based modelling to boost the security level when compared with different existing block cipher-based algorithms using the Anti-Codify Technique (ACoT).

The main contribution of the work involves the following:

- The authors used DeoxyriboNucleic Acid (DNA) model for the generation of cipher text.
- Block-cipher-based encryption is used by Anti-Codify Technique (ACoT) but the original encrypted file and its extension are broken up into separate blocks. When the original file is broken up into two separate blocks, it raises the security level and makes it more difficult for outsiders to access the data from the cloud.
- Finally, fuzzy based classification is conducted to check if the features extracted are of secured class type, such that the data stored is of secured one.

2 Related Works

In the literature, a great deal of the significance in developing the models or framework for the authenticity of data stored in cloud with accurate integrity and validity has been highlighted by a large number of researchers.

According to Takabi et al. [10] a data centre security architecture for virtual machine discs was developed that featured the use of block cipher-based encryption and decryption, as well as the usage of Merkle one-time signature scheme, to verify that the data was not corrupted. In comparison to the signature systems that were previously in place, this one is more secure.

Wang et al. [11,12] also examined the security vulnerabilities associated with virtual machine. A technique known as Cloud Visor [13] uses AES-CBC with MHT and MD5 hashing functions to guarantee enough security and integrity for cloud computing environments. The AES-CBC [14] 128-bit differential cryptanalysis, on the other hand, poses a security risk.

Wang et al. [15] proposed a comparable PoR in which the MHT was signed at the root (R) to allow for quick update and correct integrity. They used a BLS signature approach and a PoR based on the conventional Merkle hash tree to achieve their results. RSA-based signatures are both slower and less secure than the secure short signature developed by Boneh et al. [16]. Using his expertise as a third-party auditor (TPA), Chris [17] has presented a revolutionary dynamic auditing approach for verifying the authenticity of user data stored in cloud data centres. However, because of the TPA direct engagement in security inspections, it is possible that sensitive information will be leaked. Prabhakaran [18] discovered that existing cloud storage solutions made it possible for attackers to easily modify the data stored inside. They presented an upgraded group-based proof of storage technique for Internet of Things-based cloud storage. Researchers were also provided with information on how to encode and decode data when downloading [19], as well as how to ensure total retrievability. The adaptable solution developed by Lee et al. [20] can be applied on multiple IoT devices with variable memory sizes. In addition, the DNA sequence is used to generate random encryption keys that hackers find difficult to decipher. FlexCrypt, an automated, lightweight encryption system for WSNs. Clustering techniques that support the mobility of sensor nodes have been developed in the FlexCrypt scheme. The secure transmission of data and keys among the various WSN nodes will be aided by a new lightweight key management and authentication method.

It is proposed that an encryption and decryption at the cloud data centre corresponds to the aforementioned security constraints: it is an encryption and decryption method based on the e-stream cipher and it is a secure encryption and decryption method.

3 Proposed Method

The efficient structure presented in this study has made it possible to increase the security of data storage and transfer. Fig. 1 depicts an illustration of a secure cloud storage system. With this strategy, the owner data will be safely saved on multiple cloud storage servers that have been registered with the cloud service provider according to the owner (CSP).

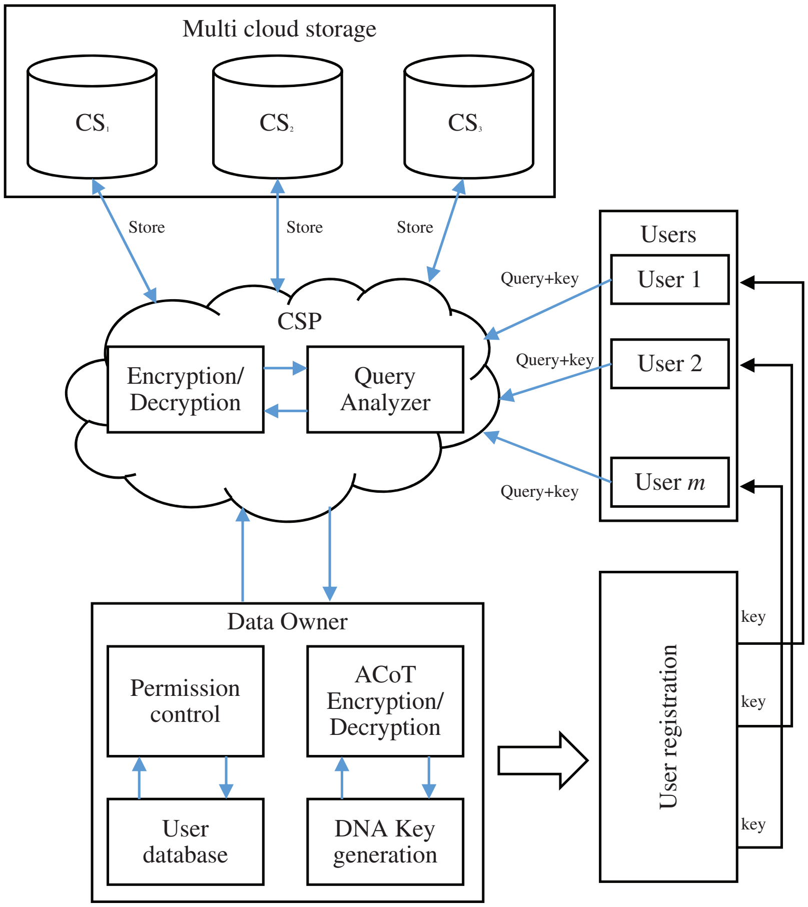

Figure 1: Proposed data storage security model

The proposed technique includes components such as DNA-based key generation, ACoT Encryption, selection of storage server and data storage, and decryption.

3.1 Key Generation

The fundamental key generating process in the first development phase has been shown in Fig. 2.

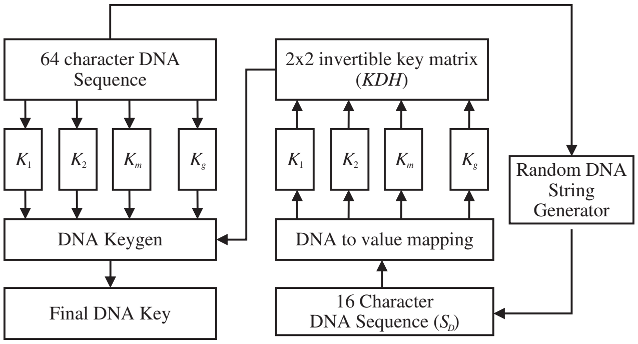
64 character DNA Sequence
2x2 invertible key matrix (KDH)
K_1
K_2
K_m
K_g
K_1
K_2
K_m
K_g
Random DNA String Generator
DNA Keygen
DNA to value mapping
Final DNA Key
16 Character DNA Sequence (S_D)

Figure 2: Key generation

To begin with, a random 64-nucleotide DNA string named KDNA was chosen at random from a large pool of candidates. In addition, the 16-nucleotide DNA sequence *SD* was chosen at random to go along with this project. Because of this, the *SD* was broken into four equal portions, each of which had four distinct DNA base combinations. The binary value Ni has been allocated to each of the SD_i(00, 01, T, 10, and C) in the following order:

Ni (A = 00, T = 01, C = 10, G = 11).

In order to use the values assigned to each Ni, a 2 × 2 matrix KDH invertible modulo with entries ranging from 0 to 250 was constructed. Mathematics can be used to construct modulo p invertible two-by-two matrices of the type $(p^2\text{-}1)*(p^2\text{-}p)$, in which each element can be selected from the set (p-1) of numbers 0–7, as well as the numbers 0–2. Following that, the recommended DNA-based key generation approach was applied to the key matrix SDH and every part of the DNA string, and the resulting K_{Ei}string was computed in the same manner as previously described.

Algorithm 1: Key Generation Process

Input: Sequence of DNA (2 Nos) ofnucleotides length (of 64 and 16).

Output: Key

Step 1: Enable the selection of 64 nucleotide sequence of a 128-bit DNA

Step 2: Split the sequence into 8 parts K_1,…K_8 each are of 16 bits.

Step 3: The individual K_i's are split into equal 8 bits and it is then considered as a column vector by appending padding.

Step 4: Generate 16 characters in random way and this is regarded as DNA sequence (SD).

Step 5: Against divide each SD (SD_1, SD_2, SD_3 and SD_4) into four parts.

Step 6: Map the sequence SD_i into binary values namely N_1, N_2, N_3, and N_4.

Step 7: Generate an invertible matrix KDH of size 2 × 2 with N_1, N_2, N_3, and N_4.

By combining all of the KE_i, we were able to construct the final 128-bit DNA encrypted key, which we designated K_f. This key will be used later on, when the communication is encrypted, to decrypt the message.

3.2 Encryption

In secret key encryption, encryption and decryption is made with the help of same key. Thus, the key used at the time of encryption; that same key is only used for the decryption, without key end-users cannot read the original text.

3.2.1 Stream Cipher

In Block cipher, the original text is treated as a block and it produce a cipher block of equal text length (See Fig. 3). It can use 64 (or) 128 bit as block size.

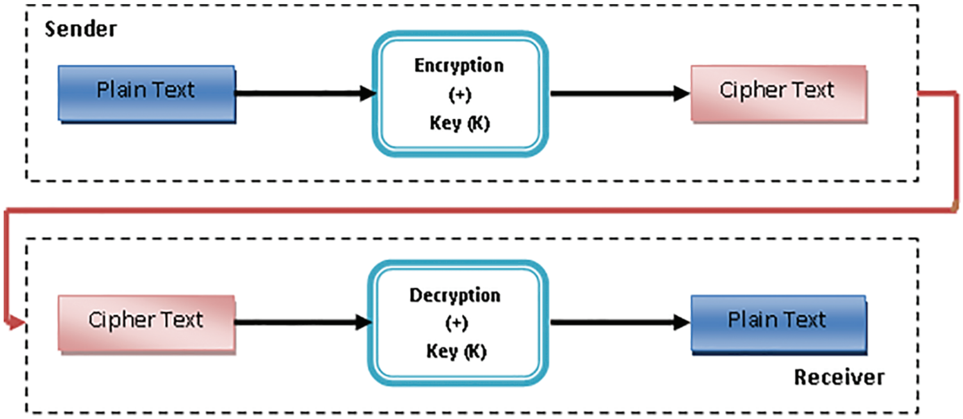

Figure 3: Block cipher process

Algorithm for Encryption on Block Cipher Operation

Step 1: Prepare subkeys for each generated DNA key

Step 2: XOR operation is performed on subkey as an initial round.

Step 3: Encryption cycles depend upon the key length of the secrete key.

a) a. 9 cycles are repeated for 128-bit key length.
b) b. 11 cycles are repeated for 192-bit key length.
c) c. 13 cycles are repeated for 256-bit key length.

Step 4: Encryption Round.

a) Each byte of the original text is converted to matrix is said to be SB (Substitute Byte) Operation.
b) First row of the matrix is shifted to the left of the matrix and second row of bytes is shifted by on position and the remaining rows are shifted as well, this operation is said to be SR (Shift Row) operation.
c) The constant matrix of size 4 bytes * 4 bytes columns are multiplied which said to be MC (Mix columns) operation
d) Adds XOR operation for each subkey bytes, it produce 16-byte long subkeys. This operation is said to be AR (Add Roundkey) operation.
e) Four operations such as Substitute Byte, Shift Row, Mix columns and Add Round Key operation is performed; depending upon the secrete key length used by the clients.

3.2.2 Anti-Codify Technique (ACoT)

Anti-Codify Technique (ACoT), is an advanced technique used for encryption process in the block cipher method encryption. It reduces the size of the encrypted data and increase the speed of encryption time for transferring the data. It converts the original text into cipher text on the format of block cipher method at the time converting into block cipher it split the encrypted data into two blocks original file and extension of the original file (See Fig. 4).

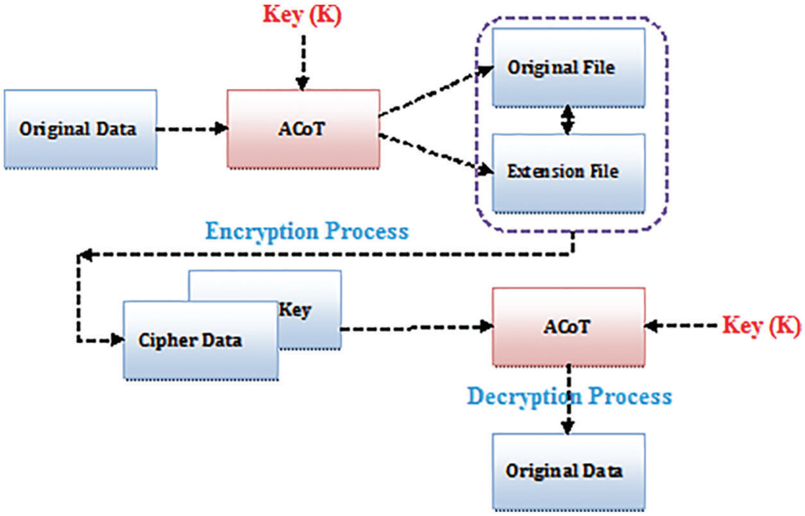

Figure 4: Anti codify technique process

Pseudo Code of ACoT:

Input: Plain_data block ($pbck_b$), Secrete key (sk)

Output: ACoT State, Plain_data

```
Begin
State = InitState($pbck_b, sk$)
AddExtension ($State, Ext, sk$)
AddKey ($State, sk$)
For ($i = 1{:}n_{r-1}$) do
    SubBytes:SB($State$)
    ShiftRows:SR($State$)
    MixColumns:MC($State$)
    AddKey ($State, Key_i$)
End For
SubBytes:SB($State$)
ShiftRows:SR($State$)
AddExtension($State, Ext$)
AddKey($State, Key(n)_{r-1}$)
End
```

ACoT split the original text into two blocks before it starts encryption. First block contains original file data that we need to store in cloud and another block contains the extension of the original file. After splitting into two different block, then encryption process is to make to convert the original data into cipher data.

ACoT split the original file data into original file block and extension file block. So the intruders cannot hack the original data easily, if the intruders find the encryption data also they cannot know the exact extension file of the data. While splitting the file it increases the encryption time and it also reduce the encryption file size. It is used to increase the speed at the time of transferring the data.

3.3 Fuzzy Based Security Classification

The data owner is in charge of decrypting the information at this phase. As soon as the data owner has successfully logged into the CSP, he or she has the ability to determine how the data is stored and safeguarded. There are three security features that will be listed and submitted to the CSP for storage: dynamic integrity (P), confidentiality (Q), and accountability (R). Security needs, on the other hand, can change these properties. These three options will be used to transmit an encrypted message, an encrypted file, and an encrypted index of the user most frequently searched terms to the specified recipients, respectively.

On the basis of the three important security characteristics outlined above, it has been demonstrated that a fuzzy-based technique may be used to store different access types (AT) of data in different cloud storage servers using a single cloud storage server. The CSP enables the data owner to choose from a number of variables when creating the CSP.

These variables will be chosen by the person who owns the data. Based on these options, a S_f will be determined using the technique described in this section. It is possible that the data owner does not understand how to apply numerical numbers to the security settings listed above. Therefore, the more vaguely defined the linguistic factors are, the better they will be in describing their significance. The CSP, for example, employs a grading range based on a language variable.

$$S = \{VP,\ VP\_P,\ P,\ P\_F,\ F,\ F\_G,\ G,\ G\_VG,\ VG\}, \tag{1}$$

where

VP = Very Poor [0 0 0 0.2],

VP_P = Very Poor to Poor [0 0.2 0.2 0.4],

P = Poor [0 0 0.2 0.4],

P_F = Poor to Fair [0 0.2 0.5 0.7],

F = Fair [0 0.3 0.5 0.7],

F_G = Fair to Good [0.3 0.5 0.8 1],

G = Good [0.6 0.8 1 1],

G_VG = Good to Very Good [0.6 0.8 1 1] and

VG = Very Good [0.8 1 1 1].

It has been accomplished here by utilising the membership function to convert the variables into ratings. The following is an example of how to define the triangle membership function using Eq. (2):

$$\mu_a(x) = \begin{cases} 0 & x \leq a_1 \\ \dfrac{x-a_1}{a_2-a_1} & a_1 \leq x \leq a_2 \\ \dfrac{x-a_3}{a_2-a_3} & a_2 \leq x \leq a_3 \\ 0 & x > a_3 \end{cases} \tag{2}$$

Following that, CSP will assess all of the term fuzzy linguistic ratings that are related to them. It was necessary to apply the graded mean and the centroid techniques in order to derive the S_f value from the fuzzy T_i value.

3.4 Storage Server Selection

Upon completion of the encryption process on the owner end, the data is transferred to the cloud and stored there. In proposed model, stored data in the cloud is stored based on geographical and independent servers that are distributed, rather than on a single server as is currently the case. With such consideration, the storage server selection phase is used to select the servers that are registered with CSP. After the data is collected, it is transferred to a cloud service provider, which subsequently divides it into smaller parts. Each data piece will be stored in a separate location due to the use of many cloud servers, security levels, and other characteristics. The distribution of the storage work over numerous cloud storage servers ensures the safe, efficient, and faster processing of the data storage task.

3.5 Calculation of the Weights of Criteria

The security of personal data is the component that causes the most concern. The time delay and processing speed are the essential features for storing the cloud data on large servers since they lower the amount of time it takes for the data to be processed. In order to assess additional communication costs associated with a cloud environment, it is necessary to know the pace at which data is transferred. The scenarios in which these attributes are required, on the other hand, can differ. Six important scale of storage server criteria were developed that is shown in Table 1 and presented in this study based on pairwise comparisons of different storage servers.

Table 1: Scale of criterion

Criterion	Significance
Intermediate	2, 4, 6, 8
Extremely important	9
Very Strongly important	7
Strongly important	5
Moderately important	3
Equally important	1

The analytic hierarchy process (AHP) methodology was utilised to obtain each criterion weights from TOPSIS method, which was then used to process the data further. The following criteria have been ranked in descending order with priority i.e., most to least significance. All of the elements that go into defining the overall level of security include the transmission speed, processing speed, time delay, and memory utilisation. When calculating the concurrency ratio (CR), we used the following formula:

$$CI/RI = 0.056 < 0.1 \tag{3}$$

As a result, the weights that were generated are reliable and can be used to guide future decision-making processes.

3.6 Storage Servers Selection on Multiple Cloud

In order of dealing with the uncertainty in selection of server, the study uses fuzzy set theory. During the selection process, a number of quantitative and qualitative aspects must be taken into consideration. As a consequence, by combining these two methodologies, the problem of storage server selection was successfully addressed. The proposed model picks the best feasible servers, where the server ratings are validated using fuzzy numbers, and this is accomplished through the use of the fuzzy TOPSIS technique.

Algorithm 3 depicts a fuzzy TOPSIS algorithm for determining the ranked server order according to their priority of selection, which is based on the values of the criteria specified above in the previous section. The distributed servers required to store the fragmented data pieces will be decided based on the quantity of the data being stored. At any given time, different chunks of the data will be saved on the storage servers with the best performance available. The last step is for the CSP to add an extra column to the data owner uploaded index IM that provides pointers to servers where bits that gets fragmented from each file is hence stored. For the purpose of safeguarding the data owner, the CSP establishes a secure link with the storage servers.

Algorithm:

Input:

weights (W) and matrix

Output: Ranked Server List

Step 1: For the purpose of converting the linguistic terminology, triangular and trapezoidal membership scales are employed.

Step 2: De-fuzzification: The graded mean is used to provide the appropriate crisp values for the triangle membership function, while the centroid method has been used to produce the appropriate crisp values for the trapezoidal membership function.

Step 3: Attribute Normalization: Following the prior stage, the values for various criteria fall into various ranges and units as a result of the preceding stage. This necessitates the employment of a normalising method to convert any attribute values that fall within the range of 0 to 1 into unit less values in order to facilitate decision-making.

Step 4: Normalization of the Higher-the-Better (HB) criterion is as follows:

$$x(i, j) = \frac{y(i, j) - \min y(i, j)}{\max y(i, j) - \min y(i, j)} \tag{4}$$

Step 5: Because of this, the Lower-the-Better (LB) criterion sequence has been normalised to the following equation:

$$x(i, j) = \frac{\max y(i, j) - y(i, j)}{\max y(i, j) - \min y(i, j)} \tag{5}$$

where

$\max y(i, j)$-highest value

(Continued)

Algorithm: (continued)

min $y(i, j)$-lowest value

x_{ij}-normalised data.

Step 6: Computation of the normalised weighted decision matrix:

$$v_{ij} = x_{ij} \times w_{ij}, \tag{6}$$

where

v_{ij}-normalised weighted data, and

w_{ij}-j^{th} criteria weight.

Step 7: Definition of the positive and negative solutions are given as below:

$$V_j^+ = \{v_1^+, v_j^+, \ldots, v_m^+\} \tag{7}$$

$$V_j^- = \{v_1^-, v_j^-, \ldots, v_m^-\}$$

Step 8: Each attribute value v_{ij} is calculated as the distance between that attribute positive ideal value (v-j) and the current value (v).

$$S_i^+ = \sqrt{\sum_{j=1}^{n} 0.3(v_{ij} - v_j^+)^2} \tag{8}$$

where $i = 1, \ldots, n; j = 1, \ldots, m$;

Step 9: To find out how far each attribute value v_{ij} is away from its negative value (v-j), perform the following computation for each attribute value:

$$S_i^- = \sqrt{\sum_{j=1}^{n} 0.3(v_{ij} - v_j^-)^2} \tag{9}$$

Step 10: Using the following formula, one can calculate how close the solution is to the ideal one:

$$CC_i = \frac{S_i^-}{S_i^+ + S_i^-} \tag{10}$$

where $CC_i = [0, 1]$;

Step 11: The storage servers are sorted according to the relative proximity coefficient, CC_i, which is expressed as a percentage.

3.7 Data Retrieval

In order to maintain data security, it is recommended that data be stored on different storage servers and that the retrieval process be carried out in the same manner. This section has been expanded with the addition of following phases:

In order to have access to the necessary files, users must first register with the CSP by creating an account with them. The user submits a registration request to the CSP, which is accepted. Requests are confirmed, and a user-friendly combination of user id and password is supplied to them, along with the unique digital signature that will be required for future access to the system in question. Each time the user wants to access any information, he or she must sign in with the signatures that have been provided. If a user username or password is incorrectly typed, the account will be temporarily suspended for a brief period of time. Because of each failed login attempt, the website response time will be a fraction of a second longer. After unsuccessful attempts, the disabling of the account takes phase for a period of time. After that length of time has passed, it will automatically restart. These strategies can be used to prevent a dictionary or brute-force assault from taking place.

3.8 Data Access Request

When a user (registered) submits a query, the query analyzer verify the index and tends to displays the encrypted files to the user that include the provided keyword as per the CSP. The registered user chooses the file(s) to be accessed and submits a request to the cloud storage provider. The CSP then forwards a request copy to the controller and then the owner provides a symmetric keyto both user and CSP.

The CSP will request and validate the key that was provided by the owner at the end user. Following the verification, the data owner is contacted by the CSP to obtain authorization. After granting the permission by the owner to the CSP to share encrypted data with the intended user, the CSP in turn defines the access type (AT) and the time limit for the session to expire in exchange for the permission.

3.9 Decryption at User End

Users are also given encrypted decryption keys by data owners over a secure channel, allowing them to decrypt the information. Because the ciphertext file is XORed with the key stream, the decryption operation of the e-stream cipher is identical to the encryption operation of the cipher. To ensure that the integrity of virtual machine discs is maintained, the Merkle B + hash tree is used. Afterwards, CSP provides an expiration time, after which it checks for an AT to grant the user data access granted by the data owner for the duration of the expiration time, after which the file is collected from a distributed storage pool and forwarded to the recipient.

4 Results and Discussions

In this section, we validate the efficacy of the entire model using Secure Cloud Simulator. The simulation is conducted on a high-end computing engine with multiple VMs from various service providers. The simulation is conducted in terms of computation time, transmission cost, average transmission speed, average processing speed, average entropy/per encryption byte, encoding and decoding inefficiency.

- Computation Time: It is defined as the total computational time in encrypting and decrypting a packet
- Transmission Cost: It is defined as the cost required to transmit an encrypted packet from source to destination host
- Average Transmission Speed: It is defined as the average speed at which the transmission takes place between the source and destination host.
- Average Processing Speed: It is defined as the speed at which the data is encrypted and transmitted.
- Average Entropy/Per Encryption Byte: It is defined as the randomness collected by a system for use in algorithms that require random datato encrypt
- Encoding Inefficiency: It is defined as the improper encryption of the data such that it gets decoded easily by the intruder
- Decoding Inefficiency: It is defined as the improper decryption of the data at the receiver end.

The results of simulation are tested against two state-of-art models that includes Merke’sone time signature and AES-CBC with MHT and MD5 hashing methods.

Fig. 5 shows the results of Computation Time between Merke'sone time signature and AES-CBC with MHT and MD5 hashing methods for Encryption and Decryption Process (ms). The results of simulation shows that the proposed method has reduced computational time in encrypting or decrypting the user data than other methods.

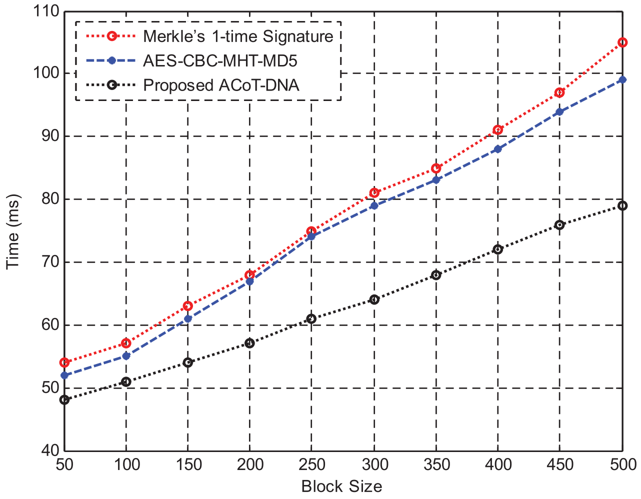

Figure 5: Computation time for encryption and decryption process (ms)

Fig. 6 shows the results of Transmission Cost between Merke'sone time signature and AES-CBC with MHT and MD5 hashing methods with time Complexity O(log n). The proposed method has reduced transmission cost while transmitting the encrypted packets from one-end to the destination.

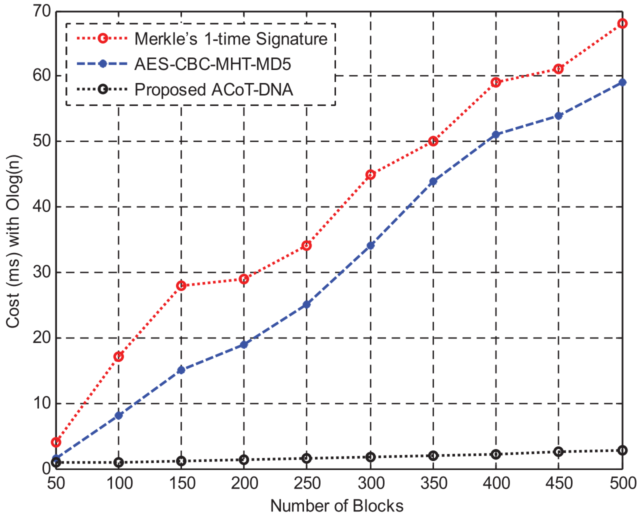

Figure 6: Transmission cost with time complexity O(log n)

Fig. 7 shows the results of Average Transmission Speed between Merke'sone time signature and AES-CBC with MHT and MD5 hashing methods. The results of simulation shows that the average transmission speed is higher in proposed system than other existing methods.

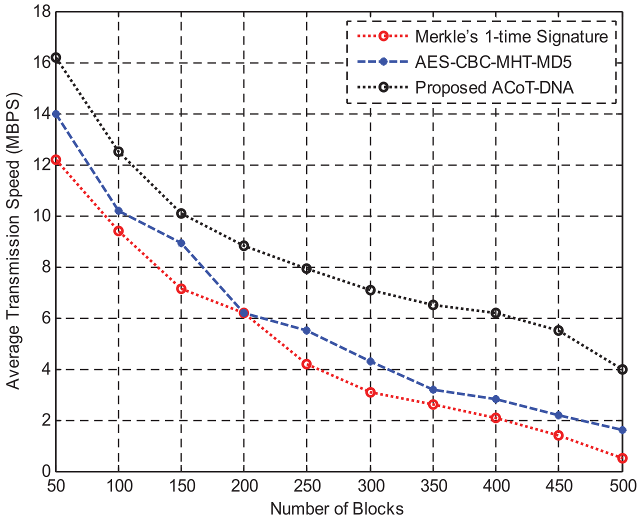

Figure 7: Average transmission speed (MBPS)

Fig. 8 shows the results of Average Processing Speed between Merke'sone time signature and AES-CBC with MHT and MD5 hashing methods. The proposed method thus achieves higher average processing speed with easier computation of keys than other methods.

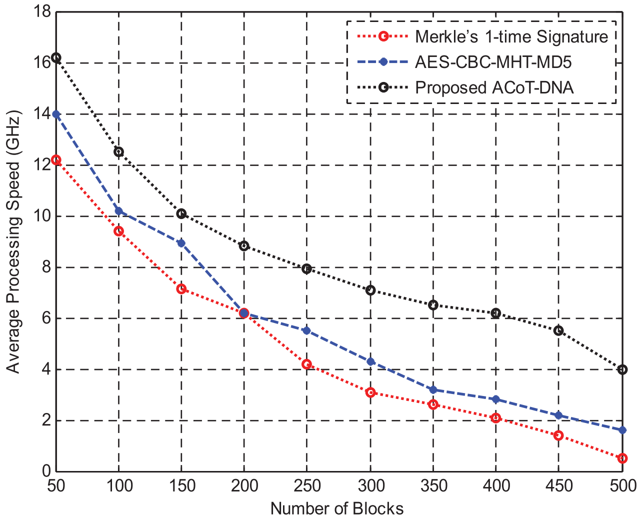

Figure 8: Average processing speed (GHz)

Fig. 9 shows the results of average entropy between Merke'sone time signature and AES-CBC with MHT and MD5 hashing methods per encryption byte. The proposed method thus achieves higher average entropy with easier computation of keys than other methods.

Fig. 10 shows the results of Encoding inefficiency between Merke'sone time signature and AES-CBC with MHT and MD5 hashing methods. The results show that the proposed method is high efficient in decoding than other methods.

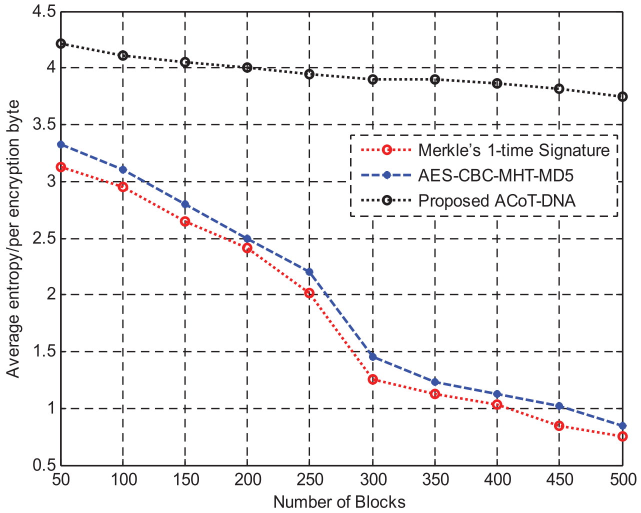

Figure 9: Average entropy/per encryption byte

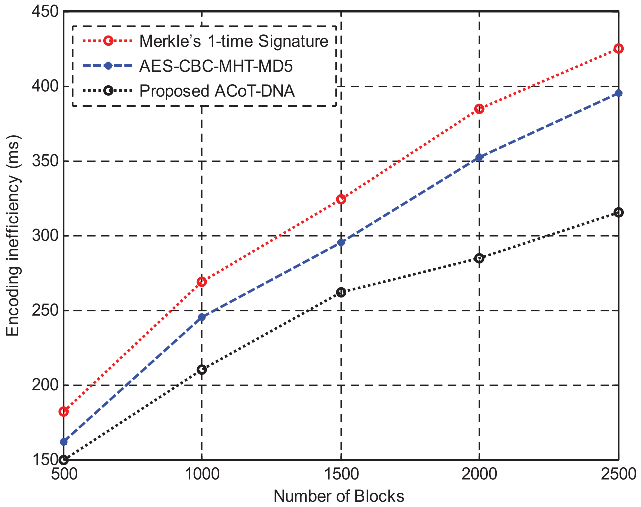

Figure 10: Encoding inefficiency

Fig. 11 shows the results of Decoding inefficiency between Merke'sone time signature and AES-CBC with MHT and MD5 hashing methods. The results of simulation shows that the decoding inefficiency is slightly lesser than encoding inefficacy. However, the results show that the proposed method is high efficient in decoding than other methods.

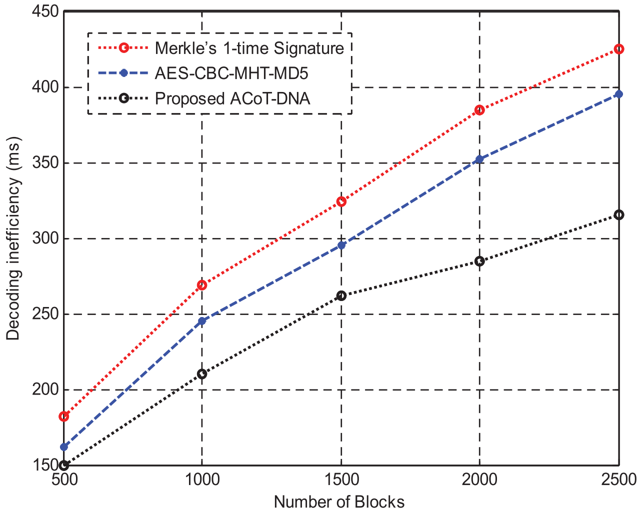

Figure 11: Decoding inefficiency

5 Conclusions

In this paper, we use a state-of-art solution to jeopardise the security risk at cloud storage solutions. The cipher file size is 35% bigger than the original file size, the encryption procedure takes significantly longer than it would otherwise. The size of the encryption file determines how long it takes to transmit a single file. Because the original files are divided up into cipher and extension files, the crypt files are smaller and transmission speeds are faster, resulting in faster transmission. As a result, the number of hackers is reduced, and the level of security in cloud storage is raised. The results further shows that the computation time is reduced in proposed method with reduced transmission cost, average transmission speed and average processing speed. The results of simulation show that the proposed method achieves higher encoding and decoding efficiency than the existing methods.

Acknowledgement: We show gratitude to anonymous referees for their useful ideas.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] K. Venkatachalam, P. Prabu, A. Almutairi and M. Abouhawwash, "Secure biometric authentication with de-duplication on distributed cloud storage," *PeerJ Computer Science*, vol. 7, no. 7, pp. 569, 2021.

[2] M. Sumathi and S. Sangeetha, "A Group-key-based sensitive attribute protection in cloud storage using modified random fibonacci cryptography," *Complex & Intelligent Systems*, vol. 7, no. 4, pp. 1733–1747, 2021.

[3] G. Hou, J. Ma, C. Liang and J. Li, "Efficient audit protocol supporting virtual nodes in cloud storage," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 5, pp. 3911, 2021.

[4] M. Anuradha, T. Jayasankar, N. B. Prakash, M. Y. Sikkandar, G. R. Hemalakshmi *et al.,* "IoT enabled cancer prediction system to enhance the authentication and security using cloud computing," *Microprocessors and Microsystems*, vol. 80, no. 103301, 2021.

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

[6] J. R. Gudeme, S. K. Pasupuleti and R. Kandukuri, "Attribute-based public integrity auditing for shared data with efficient user revocation in cloud storage," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 2, pp. 2019–2032, 2021.

[7] K. C. Purohit, M. Manchanda and A. Singh, "Cloud data storage security: The challenges and a countermeasure," *Soft Computing: Theories and Applications*, vol. 1380, pp. 97–105, 2022.

[8] T. Jiang, W. Meng, X. Yuan, L. Wang, J. Ge *et al.,* "ReliableBox: Secure and verifiable cloud storage with location-aware backup," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 12, pp. 2996–3010, 2021.

[9] A. Mohiyuddin, A. R. Javed, C. Chakraborty, M. Rizwan, M. Shabbir *et al.,* "Secure cloud storage for medical IoT data using adaptive neuro-fuzzy inference system," *International Journal of Fuzzy Systems*, vol. 24, no. 2, pp. 1203–1215, 2022.

[10] H. Takabi, J. B. Joshi and G. J. Ahn, "Security and privacy challenges in cloud computing environments," *IEEE Security & Privacy*, vol. 8, no. 6, pp. 24–31, 2010.

[11] Q. Liang, Y. Z. Wang and Y. H. Zhang, "Resource virtualization model using hybrid-graph representation and converging algorithm for cloud computing," *International Journal of Automation and Computing*, vol. 10, no. 6, pp. 597–606, 2013.

[12] B. Shao and Y. Ji, "Efficient TPA-based auditing scheme for secure cloud storage," *Cluster Computing*, vol. 24, no. 3, pp. 1989–2000, 2021.

[13] F. Zhang, J. Chen, H. Chen and B. Zang, "Cloudvisor: Retrofitting protection of virtual machines in multi-tenant cloud with nested virtualization," in *Proc. of the Twenty-Third ACM Symp. on Operating Systems Principles*, Cascais Portugal, pp. 203–216, 2011.

[14] H. Shacham and B. Waters, "Compact proofs of retrievability," in *Proc. Int. Conf. on the Theory and Application of Cryptology and Information Security (ASIACRYPT)*, Melbourne, Australia, pp. 90–107, 2008.

[15] Q. Wang, C. Wang, J. Li, K. Ren and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in *Computer Security–ESORICS 2009: 14th European Symp. on Research in Computer Security*, Saint-Malo, France, September 21–23, pp. 355–370, 2009.

[16] D. Boneh, B. Lynn and H. Shacham, "Short signatures from the weil pairing," in *Proc. Int. Conf. on the Theory and Application of Cryptology and Information Security*, Brisbane, Australia, pp. 514–532, 2001.

[17] C. C. Erway, A. Kupcu, C. Papamanthou and R. Tamassia, "Dynamic provable data possession," *ACM Transactions on Information and System Security (TISSEC)*, vol. 17, no. 4, pp. 1–29, 2015.

[18] V. Prabhakaran and A. Kulandasamy, "Hybrid semantic deep learning architecture and optimal advanced encryption standard key management scheme for secure cloud storage and intrusion detection," *Neural Computing and Applications*, vol. 33, no. 21, pp. 14459–14479, 2021.

[19] N. Munir, M. Khan and I. Hussain, "Cryptanalysis of internet of health things encryption scheme based on chaotic maps," *IEEE Access*, vol. 9, pp. 105678–105685, 2021.

[20] W. Lee and K. B. Sim, "Design and hardware implementation of a simplified DAG-based blockchain and new AES-CBC algorithm for IoT security," *Electronics*, vol. 10, no. 9, pp. 1–20, 2021.