AN EFFICIENT INTEGRITY BASED MULTI-USER CLOUD ACCESS CONTROL FRAMEWORK FOR HETEROGENEOUS CLOUD DATASETS

MOHD ANWAR ALI1, NAGESH VADAPARTHI2, L SUMALATHA3
1Research Scholar, Department of CSE, JNTUK Kakinada, India.
Assistant Professor, Department of IT, MLR Institute of Technology, Hyderabad, India.
2Professor, Department of IT, MVGR College of Engineering, Vizianagaram, India.
3Professor, Department of CSE, JNTUK Kakinada, India.
E-Mail: 1dr.mdanwarali23@gmail.com, 2itsnageshv@gmail.com, 3lsumalatha@jntucek.ac.in

ABSTRACT

The greater part of the customary cloud based applications are shaky and hard to figure the information honesty with variable hash size on heterogeneous datasets. Due to structured data and computational memory, cloud storage systems are also independent of integrity computational and data security. As the size of the cloud information records are expanding in the general population and confidential cloud servers, it is hard to encode and translate the huge information because of document configuration and restricted respectability key size. The computational time and extra room of the ordinary quality based encryption and unscrambling models are high during the information honesty check and restricted information size. For strong data encryption and decryption, a hybrid variable-sized data integrity algorithm is implemented on heterogeneous cloud data files in this paper. For improved cloud data security, this work proposes an optimized attribute-based encryption and decryption procedure for large data files. On cloud heterogeneous data types, proposed framework outperforms conventional cloud security frameworks in terms of optimization, as demonstrated by the results of the experiments.

Keywords: Cloud Data Security, Integrity, Encryption, Attribute Based Encryption.

1. INTRODUCTION

Traditional security methods and methodologies are difficult to detect the path planning security of cloud environment due to the rapid growth of computer network architecture and communication technologies. Traditional cryptography systems rely on mathematical principles as well as unproven computing limitations in order to function. The methods listed above are commonly used in applications involving secret message sharing across an insecure channel. The problem of key distribution is a serious challenge in traditional cryptography techniques. The Cloud security system records, monitors, and analyses network activity to identify whether permitted systems are being attacked. Both the misuse and anomalous detection models require network log data as training data, and the quality of this data will have a significant impact on the Cloud security and prevention system's efficiency [1]. With the increasing numbers of cloud users, they are capable of retrieving information related to their location all the time. Recently, the infrastructures of attribute based encryption schemes are unable to satisfy the requirements of location attributes. To decrease the unnecessary quantity of capital expenditures, most real-time apps and information are moved to the cloud setting. Therefore, in a very short time, cloud computing technology has gained enormous popularity. Data safety and information protection are regarded as two of cloud environments ' main issues. In order to achieve this, many efforts have been made in this field to develop an extended and advanced security algorithm. There are some important requirements of cloud computing which are listed below:-

1.1. Data Confidentiality

Data confidentiality can be defined as the limitation of the unauthorized user from accessing sensitive confidential data on cloud. It is not permitted for unlawful consumers to access information from cloud data storage. In the transmission phase, the exchanged information is always regarded confidential. Only authorized and valid users are allowed to access their confidential private data. No one other than authorized users is permitted to access these data stored on cloud. There are two types of techniques in order to
achieve data confidentiality, those are:- physical isolation and cryptography. All users’ data are needed to be encrypted properly by a strong and efficient encryption algorithm before uploading them on cloud data storage.

1.2. Integrity

The integrity of the data is compromised, when the data is changed illegally by an intruder. The intruder modifies data before it reaches to its intended destination. An alteration of sensitive data checks changes made by unauthorized user. Through the above process, the data can be protected from malicious modification. Some other approaches are also there for ensuring data integrity such as:- RAID-like techniques, hashing approaches, message authentication codes and so on.

1.3. Scalability and Efficiency

There are huge numbers of users interacting in cloud environment. Some users use cloud for short period of time, but some others use cloud for extended period. It is totally unpredictable to determine how many users can join and leave the cloud system. There, the cloud system and its architecture are required to be scalable and efficient in nature.

1.4. Availability

Cloud users can use cloud services on the Internet at anytime and anywhere. It therefore makes full use of the nature of cloud accessibility. Hardening and redundancy can be seen as two separate methods to improve cloud service accessibility. As multiple messages must be verified in a short amount of time before new messages are received, the message verification policies should be selected appropriately to improve the cloud data message verification process. To address the difficulty of fine-grained access control mechanisms, data access rights are applied in a hierarchical manner. Cloud client's information is encoded into figure text utilizing trustworthiness procedures. The decryption algorithm and the values of the decryption keys can only be accessed by approved servers. The decoding method doesn't permit an assailant to partake in the unscrambling system since it needs unscrambling keys. Numerous hardships emerge in the encryption cycle, including the expense of bilinear matching and the more slow handling velocity of asset obliged gadgets. Using virtual machine-based encryption, cloud environment data privacy and security can be significantly enhanced [2]. It permits each authorized user to define its own access integrity policies for the encryption process. Access strategies are responsible for keeping up with and controlling every client's entrance honors so that endorsed information can be gotten to. Traditional security methods and methodologies are difficult to determine the path planning security of cloud environment due to the rapid growth of computer network architecture and communication technologies. Traditional cryptography systems rely on mathematical principles as well as unproven computing limitations in order to function. The methods listed above are commonly used in applications that require secret message sharing across an unsecure medium. The difficulty of key distribution is the fundamental problem with typical cryptographic algorithms. In the Symmetric Key Cryptosystem, both encryption and decryption require the same key. However, the Asymmetric Key Cryptosystem requires two keys, one for encryption and the other for decryption. Other issues include a lack of Cloud security, complicated software and hardware infrastructure, and a lack of network monitoring, among others. The Cloud security system records, monitors, and analyses network activity to identify whether permitted systems are being attacked. Both the misuse and anomalous detection models require network log data as training data, the quality of which will have a significant impact on the Cloud security and prevention system's efficiency [3]. Traditional CP-ABE approaches are incapable of meeting the demands of scaled media sharing. To address this issue, the MCP-ABE technique is presented, which is capable of supporting scalable media. A cypher text message is created by encoding multiple messages using an encryption technique. Content delivery systems are the most extensively used application of this method. A key graph is created based on the user's access privileges, and the media units are encrypted using the generated keys. The decryption algorithm is carried out if the user possesses the necessary attributes that correspond to user access rights. Yet, in the event of cell phones, the issue emerges in view of restricted assets. The above issue is settled through the re-appropriating cycle of computational cycles to cloud climate servers. The re-evaluating system improves the productivity alongside taking care of protection and honesty of information. The process of encryption has the responsibility to limit the access in order to process cipher texts. Homomorphic encryption has the objective to assist the complete computation process. It is also used to enforce confidentiality and data integrity restrictions. The significant downside of the Homomorphic encryption
approach is the single client based framework. The regular Homomorphic encryption method never upholds various clients. Crypttexts can only perform a limited number of operations. Completely Homomorphic based CP-ABE model is created to help various calculations of code texts. On the other hand, it significantly raises the overall computation overhead. Subsequently, this strategy can't be carried out if there should be an occurrence of true applications [4]. There are a few extraordinary sorts of records which can be decoded at just specific time. The geological area of cloud clients likewise differs regularly every once in a while. Using bilinear maps is necessary for the development of an attribute-based encryption method. At the point when the quantities of traits in the entrance structure increments, then the quantity of bilinear tasks likewise increments. The majority of cases like the one above come up during the decryption process. Subsequently, it is a lot of important to diminish the decoding handling cost. It also has a responsibility to offer clients fine-grained access control [5]. Distributed computing method is respected to be the best innovation that can be participated in the data figuring framework. In many sorts of assaults and dangers, distributed computing is powerless. Since there are numerous wellbeing issues in distributed computing, the dangers and weaknesses are being explored. The exchange of resources between multiple clients is essential to cloud computing. There are several current cloud computing key management methods. The unified key management system is used for centralized main creation, modification and deletion. Several methods and techniques like TLS provide safety between the centralized main manager and the consumers. The existing methods suffer from latency and if the centralized server is compromised, then the security is lost. In the distributed key management scheme the keys are locally managed by the users, but it suffers from the overhead. In the hybrid key management schemes, it uses the combination of the two schemes, namely the centralized and distributed key management techniques [6]. The KP-ABE and the CP-ABE are used for the exchanging the keys among the users in the cloud environment. In both these schemes, if the central manager having the keys is compromised, then the security issues are violated. Hence, to increase the security issues in the cloud, here in this research the basics of the cloud computing, such as the security challenges, basics of the cloud computing are surveyed. The techniques in the existing systems are studied to maximize the security in the cloud. The cryptographic solutions are provided in the cloud to overcome the security issues [7]. The basics and the types of cryptography are studied to provide the solutions to the cloud security. The cryptographic solutions include the encryption and decryption techniques. The key distribution policies are used to overcome the drawbacks of the existing cloud systems [8].

2. RELATED WORKS

One essential cryptographic method for calculating the integrity value in message authentication and group communication is the integrity verification function. The behaviour of the virtual machine and its effects on the communication data are checked using integrity verification. For the purpose of the source verification process, a cryptographic hash function takes data of variable length and generates a digest of fixed length. The greater part of the customary trustworthiness works, for example, MD5, Whirlpool, SHA512 and turbulent hash maps are utilized to really take a look at the honesty of the text or record in the verification conventions. In both static and dynamic cloud networks, traditional integrity verification algorithms have difficulty preventing attacks and collisions [9][10]. Additionally, these algorithms are inefficient when used with large amounts of data in a dynamic cloud environment. Cryptographic integrity functions have been used in numerous research projects to verify communication data in cloud networks. During the data communication process, each cloud server is validated using the authentication model. To authenticate each cloud user in the cloud network, traditional authentication models [11] have been proposed in the literature. Essentially, these verification models [12] are tedious and infeasible for enormous cloud organizations. The following are the main issues with these models: restricted information size, challenging to create variable size respectability esteem and hard to produce a unique hash esteem on huge size cloud organizations. Information encryption, homomorphic encryption, and mystery sharing calculations were recorded as the strategies broadly utilized for protecting data re-evaluating. The paper referenced why multi-mists or between mists are liked over single mists in light of the fact that a solitary cloud struggles from a few wellbeing issues like seller secure in, information accessibility, and so forth. For multi-cloud security, the paper used a Shamir secret sharing scheme [13]. They discussed a number of issues with homomorphic encryption and explained how it was determined to be suitable
and security were chosen as the primary obstacles and serve as a promising platform for the next exploration trends were mentioned [18]. [19]. In the context of data integrity patterns, future prevalent data integrity structures was carried out. characteristics, a relative examination of the most were talked about. Based on the known evaluation the various schemes' efficiency as a parameters were mentioned for the purpose of identified and their impact was analysed. These security attacks and ways to stop them were talked about. Based on the known parameters were mentioned for the purpose of evaluating the various schemes' efficiency as a whole. The security attacks and ways to stop them were talked about. Based on the known characteristics, a relative examination of the most prevalent data integrity structures was carried out. In the context of data integrity patterns, future exploration trends were mentioned [18]. [19]. Predicted that cloud computing would reshape IT and serve as a promising platform for the next generation of the internet. Nevertheless, privacy and security were chosen as the primary obstacles to widespread adoption of cloud computing. The accuracy of information putting away and computation were uncovered to be surrendered owed to the lack of control of information security for clients. The author anticipated that "SecCloud" would be an efficient and competent cloud security protocol. The upcoming dimension remained deliberate and emphasized privacy concerns [20]. [21] Emphasized the utilization of big data within healthcare organizations. In a data multi-cloud environment, the paper described a method for the safe and private sharing of medical big data among organizations. Because of segmented information spread across multiple clouds, the architecture employed an attribute-based encryption scheme for user authorization and secret sharing. Encoded medical data was transferred and recovered simultaneously from multiple clouds using multcloud proxies. ABE and role-based access policies were used to select characteristics of a medical record. The useful viability and acceptable performance were demonstrated by a discussion of the execution and evaluation by numerous experiments. The approaching exertion was referenced to check between authoritative aspects of key administration and Job Based Admittance Control (RBAC) procedure organization, and various enhancements for the multi-cloud intermediary [22]. [23] Referenced security and key administration as key worries in distributed storage regardless of their appealing highlights. As a means of applying the secret-sharing method directly to the files for storing multiple segments of a file in multiple clouds, a plan known as "CloudStash" was proposed, implemented, and evaluated. The referenced method alongside multithreading was said to improve classification, accessibility, execution, and adaptation to internal failure. The objective of the method was to speed up upload and download speeds for medium and small files. For high-performance upload and download operations that make use of multi-threading, the design and algorithm analysis were carried out. An examination between the pattern calculation utilizing AES, SHA512 and RSA 1024 with Cloud Stash arrangement was finished. The share was signed with RSA, and each share was hashed with SHA512. A capable record level of leadership quality based encryption plot was presented in circulated figuring. The experiments were carried out in Python with the Amazon AWS S3 API and utilized numerous files and eight storage centres of AWS S3 [24, 25]. After combining the various levelled reports with the organized access structure, the layered access structures were combined into a
single access structure. The archives might share
the text fragments in the cipher that are related to
characteristics. Encryption time and cipher text
accumulation were both reduced in this manner.
Furthermore, the standard assumption held that the
presented arrangement would ultimately be secure.
The preliminary test demonstrates that the proposed
arrangement was particularly effective for
translating and encrypting data. With how much the
reports expanding, the upsides of our plan become
logically plainly obvious [26]. [27], presented a
typical structure for public key administration and
multi-access control strategies to guarantee
adaptable and fine-grained data re-evaluating
verification in a double proprietor climate using
Public Key Framework (PKI) and access control.
Key administration intricacy was limited and the
keys were disseminated utilizing a PKI-based key
management protocol.

3. MULTI-USER CLOUD SECURITY
SYSTEM FOR HETEROGENEOUS
DATASETS

Multi-client based encryption is one of the
most mind-blowing versatile methodology for
cloud information security with variable ascribes
and strategies. This technique is called ABE.
During the encryption and decryption processes,
each cloud user has a privacy key and a number of
approved Multi-user sets, policies, and encryption
keys. For keyword-based cloud data security,
umerous attribute-based encryption systems have
been used in the literature. The data size of the
multi-user is exponentially limited to security
measures because of the size of the multi-users and
the multi-user key setup phase. The access tree
structure decrypts secret data using the private key
and cipher text in a fundamental ABE operation.
The concept of CP-ABE and the inverse process of
KP-ABE are similar. CP-ABE is a fundamental unit
for numerous other extensions due to its
adaptability. As a result, users can choose one
attribute or a set of attributes from the specified set.
Hierarchical structure was used to represent the
entire technique. Job based admittance control
strategies appoints the client access control
authorizations and jobs as per the business
capability in the association. The relationship
between the user and the access permissions is the
role. The general system of the proposed model is
addressed in figure 1. In the figure 1, client's
feedback information is taken as information and
trustworthiness esteem is registered to every client's
quality for the property based encryption process of CP-ABE conspire.

In the figure 1, a hybrid integrity algorithm is
proposed using the non-linear chaotic function. In
this integrity approach, user’s attributes are taken as
input for the variable hash value generation. Each
cloud user’s input records and integrity value are
used to encrypt the data using the proposed attribute
based encryption scheme. This work is
implemented two phases. In the main stage, a half
and half non-straight trustworthiness approach is
executed on the multi-client's credits. In the
subsequent stage, a half and half trustworthiness
based CPABE conspire is proposed for the
information encryption and decoding process.

3.1 Phase 1: Non-linear hash key generator

In this phase, each user’s attribute are taken as
input for the hash value as unique identity. This
unique identity is used to construct policies for the
attribute based encryption and decryption process.
The basic steps in the non-linear integrity
computation are presented below.

I: Initialization user’s attribute A[], message
M_data . For each attribute in the A[]
Do
1f(M_data!=""")
them
M_Byte=Bytes(A[i]);
Done
2: Partition the input data $M = M_{\text{bytes}}$ into $k$ blocks.
3: Input message is padded if the length exceed the block size with 0000001
4: To each block in the $k$ blocks $B[k]$.
   Partition each block into sub-blocks of size 32bits. These sub-blocks are used to perform a sequence of non-linear transformations $\text{Subblock partitions } S_{\text{P}[i]} = B[S/32]$;
   To each sub block in the $S_{\text{P}[i]}$
   Perform Non-linear transformation function (SP[i])
   Done
5: Perform non-linear sequence of transformation as nonlinearT
   For each input byte in $S_{\text{Bytes[i]} \text{(sub block partition byte array)}}$
   Perform block operation.

In the proposed model, a novel integrity based cloud data security framework is proposed for the hash mechanism. In the traditional cloud-based hash security framework, traditional integrity algorithms such as MD5, SHA, Whirlpool etc are utilized for checking the integrity of data in the cloud computing environment. In the proposed framework, a hybrid non-linear dynamic integrity algorithm has been proposed to improve efficiency of hash bit variation in the hash construction process.

3.2 Non-linear transformation function
In this non-linear chaotic function, $Q$ and $R$ represent the dynamic permutation matrices. These matrices are generate using the PNLCF function.

   For each byte in P[i]
   Do
   \[ R_1 = SK^T \cdot [R, \text{MaxEigen}(SK), (\text{MaxCoefficient}(\text{Poly}(SK))) ] \]
   \[ R_2 = \left( \frac{\text{SumOfSquares}(SK) \cdot \text{det}(SK)}{\sum SK[i]} \right) \]
   \[ R_3 = \sum SS(Q) \cdot \text{Eigen}(Q,R) \]
   \[ H[i] = R_1 \oplus R_2 \oplus R_3 \]
   Done

Description: In the step 1, input data is converted in to byte array using the cloud user ID as $S$ _id and its corresponding record as $S$.data. This step is repeated to each multi-user data in the given transactions list. In the step 2, input data $M$ is partitioned into $k$ blocks with each size 8bits. In the step 3, padding operation is performed on the input data if the message size exceeds the block size. In the step 4, each block in the $k$ blocks is partitioned into subblocks of each 32bits. In the step 5, a sequence of mathematical transformations is applied on the subblock partition for hash computation. In the step 6, all the subblock hash values are concatenated as final hash value.

3.3 Phase 2: Multi-user Encryption and decryption approach
In the context of a real-time cloud computing environment, the Multi-authority ABE approach was developed. As per this methodology, various gatherings are answerable for conveyance of client credits. Multi-authority ABE method approach relies upon $K$ quantities of Multi-client specialists and a solitary focal power. A value $d_k$ is associated with every Multi-user. The fundamental stages in the MA-ABE technique are:

In this step, every client's credits and their entrance arrangements are utilized to figure the expert key and public key for the cloud based information security in the hash structure. A randomized hash key-based policy for the key generation process is built in this step.

The cyclic group elements for the cloud security initialization procedure are $G_1$, $G_2$, and $Z_p$.

The randomized cyclic group elements are used to generate a hybrid master-key and public-key based on a multi-user policy during this setup procedure. The bilinear pairing elements serve as the foundation for the construction of the setup process's multi-user master and public key components.

Geodist(x) = \((1-x)^p\), \( p = 0,1,2,\ldots \)

UniDist(m) = \(m/(d_1-d_2)\) for \(d_1 \leq m \leq d_2\)
Let $Z_r, G_1, G_2$ are multi-user access control based cyclic group elements.

$\alpha = \text{bilinear_map}(Z_r, \mu_{\text{GeoDist}()})$;

$\text{Mult_PubK}(g) = \text{bilinear_map}(G_1(), \text{max} \{\mu_{\text{UniDist}()}, \mu_{\text{GeoDist}()})$;

$\text{Mult_PubK}(\text{gp}) = \text{bilinear_map}(G_2(), \sigma_{\text{GeoDist}()})$;

$\text{Multi_Mask}(\beta) = \text{bilinear_map}(G_2(), \sigma_{\text{UniDist}()})$;
4. EXPERIMENTAL RESULTS

Exploratory outcomes are executed continuously cloud server with java climate. The hash framework is implemented in this work with multi-user data and a real-time Amazon AWS server. Integrity and security algorithms are implemented using a variety of third-party libraries in the work, including apache math, JAMA, java pairing, and the Amazon Web Services JDK. On the cloud transactions data, the experiment evaluation computes the hash bit change, hash runtime (ms), cloud encryption runtime (ms), and decryption runtime (ms). The measurement of the impact that altering input data bits has on the integrity bits is represented by the hash bit change. Traditional integrity algorithms like SHA, MD5, Whirlpool, and parallel chaotic hash were used in the experimental evaluation. Likewise, proposed encryption model is contrasted with the customary models like CP-ABE, KP-ABE, HCP-ABE and Fuzzy CP-ABE in the exploratory assessment.

Table 1, describes the secret key value of the input data which is generated in the encryption process. This key is generated using the integrity value and the attributes list.
Table 2, describes the master key value of the input data which is generated in the encryption process. This master key is generated in the setup phase of the proposed encryption model. This key is generated using the integrity value, policies list, and the attributes list.

Table 3, describes the public key value of the input data which is generated in the encryption process. This public key is generated in the setup phase of the proposed encryption model. This key is generated using the integrity value, policies list, and the attributes list.
Table 4, describes the sample encryption process of the proposed model on the medical records.

Table 5: Results On Single User Integrity Value In Block Wise Processing

| c[j] = p[j] XOR c[j-1] XOR y = 0x02941d5f | 0x2dec890b |
| r = (sum of all bytes c[j]) = 274 | r = (sum of all bytes c[j]) = 413 |
| r = RMIN + (r mod S) = 6 | r = RMIN + (r mod S) = 5 |
| 0xe4c70355 | 0x3a30c953 |
| r = (sum of all bytes c[j]) = 460 | r = (sum of all bytes c[j]) = 842 |
| r = RMIN + (r mod S) = 2 | r = RMIN + (r mod S) = 4 |
| 0x1dc2956e | 0x3f204758 |
| c[j] = p[j] XOR c[j-1] XOR y = 0x102ebec6 | c[j] = p[j] XOR c[j-1] XOR y = 0x0ede169b |
| r = (sum of all bytes c[j]) = 450 | r = (sum of all bytes c[j]) = 413 |
| r = RMIN + (r mod S) = 2 | r = RMIN + (r mod S) = 4 |
| 0x938a0782 | 0xc1b63100 |
| r = (sum of all bytes c[j]) = 450 | r = (sum of all bytes c[j]) = 842 |
| r = RMIN + (r mod S) = 2 | r = RMIN + (r mod S) = 4 |
| 0x1dc2956e | 0x3f204758 |
| r = (sum of all bytes c[j]) = 450 | r = (sum of all bytes c[j]) = 842 |
| r = RMIN + (r mod S) = 2 | r = RMIN + (r mod S) = 4 |
| 0x938a0782 | 0xc1b63100 |
| r = (sum of all bytes c[j]) = 450 | r = (sum of all bytes c[j]) = 842 |
| r = RMIN + (r mod S) = 2 | r = RMIN + (r mod S) = 4 |
| 0x938a0782 | 0xc1b63100 |
| r = (sum of all bytes c[j]) = 450 | r = (sum of all bytes c[j]) = 842 |
| r = RMIN + (r mod S) = 2 | r = RMIN + (r mod S) = 4 |
| 0x938a0782 | 0xc1b63100 |

Final hash integers:

```
0x2dec890b
```

```
c[j] = p[j] XOR c[j-1] XOR y = 0x0ede169b
```

```
r = (sum of all bytes c[j]) = 413
```

```
r = RMIN + (r mod S) = 5
```

```
0x3a30c953
```

```
c[j] = p[j] XOR c[j-1] XOR y = 0xb4efdfc8
```

```
r = (sum of all bytes c[j]) = 842
```

```
r = RMIN + (r mod S) = 4
```

```
0x3f204758
```

```
c[j] = p[j] XOR c[j-1] XOR y = 0x0ede169b
```

```
r = (sum of all bytes c[j]) = 413
```

```
r = RMIN + (r mod S) = 5
```

```
0x3a30c953
```

```
c[j] = p[j] XOR c[j-1] XOR y = 0xb4efdfc8
```

```
r = (sum of all bytes c[j]) = 842
```

```
r = RMIN + (r mod S) = 4
```

```
0x3f204758
```

```
c[j] = p[j] XOR c[j-1] XOR y = 0x0ede169b
```

```
r = (sum of all bytes c[j]) = 413
```

```
r = RMIN + (r mod S) = 5
```

```
0x3a30c953
```

```
c[j] = p[j] XOR c[j-1] XOR y = 0xb4efdfc8
```

```
r = (sum of all bytes c[j]) = 842
```

```
r = RMIN + (r mod S) = 4
```

```
0x3f204758
```

```
c[j] = p[j] XOR c[j-1] XOR y = 0x0ede169b
```

```
r = (sum of all bytes c[j]) = 413
```

```
r = RMIN + (r mod S) = 5
```

```
0x3a30c953
```

```
c[j] = p[j] XOR c[j-1] XOR y = 0xb4efdfc8
```

```
r = (sum of all bytes c[j]) = 842
```

```
r = RMIN + (r mod S) = 4
```

```
0x3f204758
```

```
c[j] = p[j] XOR c[j-1] XOR y = 0x0ede169b
```

```
r = (sum of all bytes c[j]) = 413
```

```
r = RMIN + (r mod S) = 5
```

```
0x3a30c953
```

```
c[j] = p[j] XOR c[j-1] XOR y = 0xb4efdfc8
```

```
r = (sum of all bytes c[j]) = 842
```

```
r = RMIN + (r mod S) = 4
```

```
0x3f204758
```

```
c[j] = p[j] XOR c[j-1] XOR y = 0x0ede169b
```

```
r = (sum of all bytes c[j]) = 413
```

```
r = RMIN + (r mod S) = 5
```

```
0x3a30c953
```

```
c[j] = p[j] XOR c[j-1] XOR y = 0xb4efdfc8
```

```
r = (sum of all bytes c[j]) = 842
```

```
r = RMIN + (r mod S) = 4
```

```
0x3f204758
```

```
c[j] = p[j] XOR c[j-1] XOR y = 0x0ede169b
```
Table 5, illustrates the sample integrity value of the hash framework on a single hash record.

Figure 2: Performance Of Proposed Integrity Verification Model To Existing Hash Based Approaches On Variable Size Attributes (Hash Size =2048)

Figure 2: On various multi-user records with variable size attributes, the hash bit variation of the proposed multi-hash approach to the existing integrity approaches is depicted in Figure 2. Here, it is noticed that the non-direct uprightness model has better hash bit minor departure from various multi-client records.

Figure 3: Performance Of Proposed Integrity Based Encryption Model To Existing Hash Approaches On Variable Size Attributes And Multi-User Data (Hash Size =4096)

Figure 3, illustrates the hash bit variation of the nonlinear multi-hash approach to the existing integrity approaches on different transactions with variable size attributes. In this figure, it is observed that the non-linear integrity model has much better hash bit variation on different transaction data.
Table 6: Performance Analysis Of Proposed Integrity Model And Conventional Approaches On Multi-User Data (Hash Size=2048)

<table>
<thead>
<tr>
<th>Transactions</th>
<th>SHA512</th>
<th>MD5</th>
<th>Whirlpool</th>
<th>Parallel_Chaotic</th>
<th>MultiUser-Integrity</th>
</tr>
</thead>
<tbody>
<tr>
<td>HR-1</td>
<td>5786</td>
<td>4846</td>
<td>5535</td>
<td>5355</td>
<td>4639</td>
</tr>
<tr>
<td>HR-2</td>
<td>4865</td>
<td>5620</td>
<td>5761</td>
<td>5452</td>
<td>4278</td>
</tr>
<tr>
<td>HR-3</td>
<td>5643</td>
<td>5027</td>
<td>5568</td>
<td>5566</td>
<td>4424</td>
</tr>
<tr>
<td>HR-4</td>
<td>5152</td>
<td>4988</td>
<td>5931</td>
<td>5837</td>
<td>4418</td>
</tr>
<tr>
<td>HR-5</td>
<td>5347</td>
<td>5402</td>
<td>5334</td>
<td>5383</td>
<td>4610</td>
</tr>
<tr>
<td>HR-6</td>
<td>5750</td>
<td>4922</td>
<td>5467</td>
<td>5777</td>
<td>4610</td>
</tr>
<tr>
<td>HR-7</td>
<td>5533</td>
<td>5324</td>
<td>4849</td>
<td>5452</td>
<td>4361</td>
</tr>
<tr>
<td>HR-8</td>
<td>5604</td>
<td>5723</td>
<td>5047</td>
<td>5370</td>
<td>4635</td>
</tr>
<tr>
<td>HR-9</td>
<td>5326</td>
<td>4881</td>
<td>5342</td>
<td>5387</td>
<td>4518</td>
</tr>
<tr>
<td>HR-10</td>
<td>4925</td>
<td>5592</td>
<td>5597</td>
<td>5150</td>
<td>4318</td>
</tr>
<tr>
<td>HR-11</td>
<td>5802</td>
<td>5620</td>
<td>4967</td>
<td>4870</td>
<td>4114</td>
</tr>
<tr>
<td>HR-12</td>
<td>5516</td>
<td>5242</td>
<td>5830</td>
<td>5412</td>
<td>4115</td>
</tr>
<tr>
<td>HR-13</td>
<td>4866</td>
<td>5108</td>
<td>5580</td>
<td>4940</td>
<td>4335</td>
</tr>
<tr>
<td>HR-14</td>
<td>4966</td>
<td>5385</td>
<td>5110</td>
<td>4949</td>
<td>4472</td>
</tr>
<tr>
<td>HR-15</td>
<td>5355</td>
<td>5025</td>
<td>5166</td>
<td>4982</td>
<td>4051</td>
</tr>
<tr>
<td>HR-16</td>
<td>4861</td>
<td>5028</td>
<td>4907</td>
<td>5308</td>
<td>4400</td>
</tr>
<tr>
<td>HR-17</td>
<td>5508</td>
<td>5098</td>
<td>5212</td>
<td>5320</td>
<td>4435</td>
</tr>
<tr>
<td>HR-18</td>
<td>5680</td>
<td>5403</td>
<td>5361</td>
<td>4920</td>
<td>4545</td>
</tr>
<tr>
<td>HR-19</td>
<td>5483</td>
<td>4902</td>
<td>5847</td>
<td>5790</td>
<td>4518</td>
</tr>
<tr>
<td>HR-20</td>
<td>5511</td>
<td>5630</td>
<td>5461</td>
<td>5797</td>
<td>4648</td>
</tr>
</tbody>
</table>

Table 6, represents the runtime analysis of non-linear integrity model to the conventional models on heterogeneous data. In the setup, different attributes and transactions are used to compute the runtime of each transaction.
Table 7: Performance analysis of proposed integrity model and conventional approaches on multi-user data (Hash size=4096)

<table>
<thead>
<tr>
<th>Transactions</th>
<th>SHA512</th>
<th>MD5</th>
<th>Whirlpool</th>
<th>Parallel_Chaotic</th>
<th>MultiUser-Integrity</th>
</tr>
</thead>
<tbody>
<tr>
<td>HR-1</td>
<td>5232</td>
<td>5400</td>
<td>5091</td>
<td>5644</td>
<td>4269</td>
</tr>
<tr>
<td>HR-2</td>
<td>5237</td>
<td>5312</td>
<td>5605</td>
<td>5430</td>
<td>4523</td>
</tr>
<tr>
<td>HR-3</td>
<td>5749</td>
<td>4880</td>
<td>5769</td>
<td>5659</td>
<td>4542</td>
</tr>
<tr>
<td>HR-4</td>
<td>5426</td>
<td>5729</td>
<td>4921</td>
<td>5602</td>
<td>4376</td>
</tr>
<tr>
<td>HR-5</td>
<td>5095</td>
<td>5817</td>
<td>5113</td>
<td>5013</td>
<td>4287</td>
</tr>
<tr>
<td>HR-6</td>
<td>4873</td>
<td>4999</td>
<td>4847</td>
<td>4972</td>
<td>4374</td>
</tr>
<tr>
<td>HR-7</td>
<td>4844</td>
<td>5500</td>
<td>5081</td>
<td>5410</td>
<td>4433</td>
</tr>
<tr>
<td>HR-8</td>
<td>5678</td>
<td>5149</td>
<td>5200</td>
<td>4921</td>
<td>4317</td>
</tr>
<tr>
<td>HR-9</td>
<td>5131</td>
<td>5212</td>
<td>5232</td>
<td>4916</td>
<td>4524</td>
</tr>
<tr>
<td>HR-10</td>
<td>4948</td>
<td>5469</td>
<td>4957</td>
<td>5562</td>
<td>4460</td>
</tr>
<tr>
<td>HR-11</td>
<td>5007</td>
<td>5046</td>
<td>5782</td>
<td>5063</td>
<td>4163</td>
</tr>
<tr>
<td>HR-12</td>
<td>5859</td>
<td>5649</td>
<td>4996</td>
<td>5364</td>
<td>4426</td>
</tr>
<tr>
<td>HR-13</td>
<td>5564</td>
<td>5775</td>
<td>5227</td>
<td>4991</td>
<td>4390</td>
</tr>
<tr>
<td>HR-14</td>
<td>5667</td>
<td>5349</td>
<td>5248</td>
<td>5088</td>
<td>4071</td>
</tr>
<tr>
<td>HR-15</td>
<td>5753</td>
<td>5264</td>
<td>5334</td>
<td>5867</td>
<td>4627</td>
</tr>
<tr>
<td>HR-16</td>
<td>5301</td>
<td>5206</td>
<td>5511</td>
<td>4885</td>
<td>4586</td>
</tr>
<tr>
<td>HR-17</td>
<td>4917</td>
<td>5829</td>
<td>5078</td>
<td>4868</td>
<td>4657</td>
</tr>
<tr>
<td>HR-18</td>
<td>5544</td>
<td>5481</td>
<td>5089</td>
<td>5250</td>
<td>4381</td>
</tr>
<tr>
<td>HR-19</td>
<td>5501</td>
<td>4860</td>
<td>5328</td>
<td>5777</td>
<td>4492</td>
</tr>
<tr>
<td>HR-20</td>
<td>5263</td>
<td>5412</td>
<td>5645</td>
<td>5229</td>
<td>4175</td>
</tr>
</tbody>
</table>

Table 7, represents the runtime analysis of non-linear integrity model to the conventional models on transactions data (hash size=4096). In the setup, different attributes and transactions are used to compute the runtime of each transaction.
Figure 4: Performance Of Proposed Parallel Multi-User Based Encryption Model To The Traditional Integrity Based CP-ABE Models For Encryption And Decryption Runtime.

Table 8: Performance Of Proposed Multi-User Based Multi-Modal Encryption Model To The Traditional Integrity Based CP-ABE Models For Encryption And Decryption Runtime (Ms)

<table>
<thead>
<tr>
<th>Users</th>
<th>MD5 + AES</th>
<th>MD5 + DES</th>
<th>MD5 + ECC</th>
<th>MD5 + Linear CP-ABE</th>
<th>Whirlpool + CP-ABE</th>
<th>Proposed Multi-level-ABE</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-1</td>
<td>5846</td>
<td>5889</td>
<td>6040</td>
<td>6858</td>
<td>6168</td>
<td>5060</td>
</tr>
<tr>
<td>U-2</td>
<td>6050</td>
<td>6094</td>
<td>5994</td>
<td>5952</td>
<td>6769</td>
<td>4837</td>
</tr>
<tr>
<td>U-3</td>
<td>6020</td>
<td>6730</td>
<td>6045</td>
<td>6262</td>
<td>6015</td>
<td>5064</td>
</tr>
<tr>
<td>U-4</td>
<td>6266</td>
<td>6106</td>
<td>6640</td>
<td>6010</td>
<td>6656</td>
<td>4894</td>
</tr>
<tr>
<td>U-5</td>
<td>6308</td>
<td>6410</td>
<td>6478</td>
<td>6634</td>
<td>6592</td>
<td>4989</td>
</tr>
<tr>
<td>U-6</td>
<td>6194</td>
<td>6290</td>
<td>5844</td>
<td>6103</td>
<td>6403</td>
<td>5272</td>
</tr>
<tr>
<td>U-7</td>
<td>6763</td>
<td>5874</td>
<td>5964</td>
<td>6333</td>
<td>6531</td>
<td>5155</td>
</tr>
<tr>
<td>U-8</td>
<td>6812</td>
<td>6131</td>
<td>6575</td>
<td>5960</td>
<td>6002</td>
<td>4898</td>
</tr>
<tr>
<td>U-9</td>
<td>5864</td>
<td>6197</td>
<td>5887</td>
<td>6586</td>
<td>6522</td>
<td>5041</td>
</tr>
<tr>
<td>U-10</td>
<td>6407</td>
<td>5901</td>
<td>6690</td>
<td>5919</td>
<td>6873</td>
<td>4897</td>
</tr>
<tr>
<td>U-11</td>
<td>6446</td>
<td>5861</td>
<td>6381</td>
<td>6046</td>
<td>6633</td>
<td>4895</td>
</tr>
<tr>
<td>U-12</td>
<td>6071</td>
<td>6343</td>
<td>5912</td>
<td>6028</td>
<td>6850</td>
<td>5001</td>
</tr>
<tr>
<td>U-13</td>
<td>5866</td>
<td>6687</td>
<td>6695</td>
<td>6006</td>
<td>5995</td>
<td>5023</td>
</tr>
<tr>
<td>U-14</td>
<td>6188</td>
<td>6714</td>
<td>6709</td>
<td>6669</td>
<td>6635</td>
<td>5227</td>
</tr>
<tr>
<td>U-15</td>
<td>6871</td>
<td>6213</td>
<td>5840</td>
<td>5891</td>
<td>6826</td>
<td>4777</td>
</tr>
<tr>
<td>U-16</td>
<td>6191</td>
<td>6214</td>
<td>6232</td>
<td>6870</td>
<td>6516</td>
<td>5109</td>
</tr>
<tr>
<td>U-17</td>
<td>6799</td>
<td>6331</td>
<td>6156</td>
<td>5929</td>
<td>5935</td>
<td>5047</td>
</tr>
<tr>
<td>U-18</td>
<td>6036</td>
<td>5950</td>
<td>6308</td>
<td>5888</td>
<td>6445</td>
<td>5342</td>
</tr>
<tr>
<td>U-19</td>
<td>6423</td>
<td>6726</td>
<td>6024</td>
<td>6375</td>
<td>6054</td>
<td>4931</td>
</tr>
<tr>
<td>U-20</td>
<td>6330</td>
<td>6539</td>
<td>6193</td>
<td>6591</td>
<td>6803</td>
<td>5016</td>
</tr>
</tbody>
</table>

Table 8, shows the correlation between proposed structure and the current models for the encryption and unscrambling runtime (ms). The table shows that the proposed multi-modular structure has low normal runtime(s) contrasted with the customary security models.
5. CONCLUSION

On the real-time cloud computing environment, a hybrid multi-user based cloud data security framework is designed and implemented in this work. This hybrid integrity-based multi-user is intended to boost the overall efficiency of the cloud database security model. Because document searching takes a lot of memory and computational power, the majority of traditional multi-user models do not require integrity checks. In order to boost the efficiency of the runtime of integrity and encryption models in multi-user operations, this work creates a hybrid integrity model and encryption model. At last, exploratory outcomes show that the proposed multi-client advancement model has improved proficiency than the ordinary models as far as runtime and hash bit change.

REFERENCES:


