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

MOHD ANWAR ALI[1], NAGESH VADAPARTHI[2], L SUMALATHA[3]
[1]Research Scholar, Department of CSE, JNTUK Kakinada, India.
Assistant Professor, Department of IT, MLR Institute of Technology, Hyderabad,India.
[2]Professor, Department of IT, MVGR College of Engineering, Vizianagaram, India.
[3]Professor, Department of CSE, JNTUK Kakinada, India

E-Mail: [1]dr.mdanwarali23@gmail.com, [2]itsnageshv@gmail.com, [3]lsumalatha@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 for cloud storage of documents. It explained that in addition to providing data privacy during communication, homomorphic encryption had additional capabilities for calculating over encoded information, searching encrypted data, and other functions. In order to repartition the data and roughly achieve fully homomorphic encryption, a multi-cloud design with N dispersed servers was proposed for processing encrypted data [14]. The security of the framework was improved by expanding the secrecy of information and execution by dividing the put away information utilizing Information Apportioning Calculation (DPA) among various cloud providers to (1) lessen the apprehension about the data breaks and (2) increment the equal regulation executing homomorphic encryption. The upcoming effort would concentrate on putting the proposed architecture into action and conducting safety and performance tests to demonstrate its viability [15]. Liu et.al, investigated the issues, clarifications, and limitations of cloud security. The creator corresponded data classification and client validation. This paper referenced that the unwavering quality of distributed computing activities was impacted by the execution of wellbeing strategies and hence, security defects and deficiencies should be handled. The primary components of customer, connection, and server-side security, all of which operate in a shared environment; As a result, concerns about their safety and secrecy must be addressed. The availability of servers, multitenant services, data storage, access control, and identity protection of issues were discussed. Various answers for address them were proposed like homomorphic encryption, solid qualification the board, appropriated admittance control, etc., [16]. The paper emphasized the importance of information integrity systems and sought to comprehend the sanctuary concerns associated with cloud storage [17]. A set of data integrity schemes' parameters were identified and their impact was analysed. These 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 multicloud 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 exami0nation 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, numerous 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. In this structure, every client's trait respectability esteem is taken as character for the policy construction processs of CP-ABE conspire.

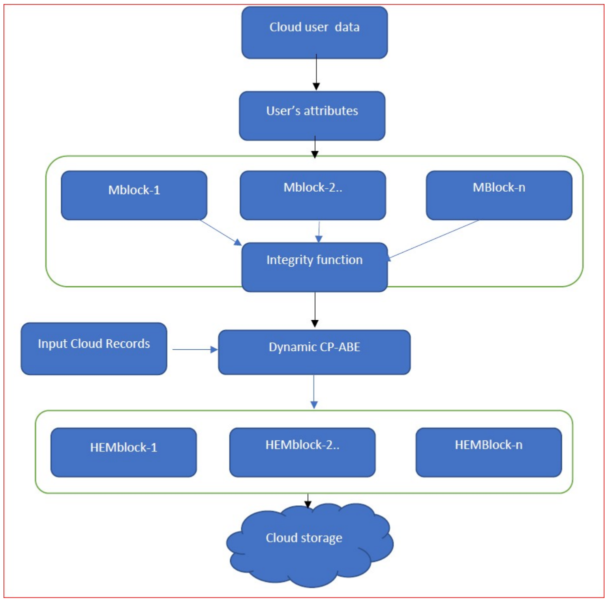

Figure 1 Hybrid Integrity Framework

In the figure1, 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.

```
1: Initialization user's attribute A[] , message
M_data .   For each attribute in the A[]
   Do
   If(M_data!==””)
   then
   M_Bytes=Bytes(A[i]);
   Done
```

```
2: Partition the input data M=M_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-lineartransformations
    Subblock partitions S_P[]=B[S/32];
    To each sub block in the S_P[]
    Do
            Perform **Non-linear transformation function** (SP[i])
     Done
5 : Perform non-linear sequence of transformation as nonlinearT
   For each input byte in S_Bytes[i]( 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
$R_1 = SK^T . [R.MaxEigen(SK).(MaxCoeffient(Poly(SK)))]$
 $R_2 = (\frac{[Q.SumofSquares(SK).det(SK)]}{(\sum SK[i])})$
 $R_3 = \sum SS(Q) * Eigen(Q.R)$
$H[i] = R_1 \oplus R_2 \oplus R_3$
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 step5, 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 dk 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 G1, G2, and Zp.

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.

$$\mathrm{GeoDist}(x) = x(1-x)^p, \quad p = 0,1,2,\ldots$$

$$\mathrm{UniDist}(m) = m/(d1-d2) \quad \text{for } d1 \le m \le d2$$

Let Zr, G1, G2 are multi-user access control based cyclic group elements.

$$\alpha = \mathrm{bilinear\_map}(Zr, \mu_{\mathrm{GeoDist}(x)});$$

$$\mathrm{Mult\_PubK}(g) = \mathrm{bilinear\_map}(G1(), \max\{\mu_{\mathrm{UniDist}(x)}, \mu_{\mathrm{GeoDist}(x)});$$

$$\mathrm{Mult\_PubK}(gp) \quad = \mathrm{bilinear\_map}(G2(), \sigma_{\mathrm{GeoDist}(x)});$$

$$\mathrm{Multi\_MasK}(\beta) \quad = \mathrm{bilinear\_map}(G2(), \sigma_{\mathrm{UniDist}(x)});$$

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: Secret Key Generated Using The Encryption Algorithm

```
E___€_<_šÔ_peÏ_X+é‘_=z˜’ò

æ‡Àyaë°÷Í GÇ__[Çƒ]À)__âÎÁ¨28—Š—[eLNDäÓs‡a

NÆˆš¤c_Qm_t+pˆ°¿Ò>mñD__sò;I† U×ÈãtÃÇA"_"Èh
_žl(ÉfÉ¡ÉþhÚm¼|pöô€q‹Ó[____S_€a6767874c74a5c
9e591622a8a04445d4853e3ecae93136faa4c4f0bcc422
f1179e41d049e2c5f276109407f5b86ee64f1f5e9cc5aba
4b1e9d884c3ef7e99e4ddE___€[øÛ[upDv•v¢ãª_  ©

¥_þÝ_9sÆ®£À__B¶Âöü  Ì™C³_tøù¸:_êÀKç:}˜¢“_ƒr
         eZV___h_Ë¤ÝÇ… o_A8_#Ï†ÏcañFãž
          )h5]vlúW ²  œ‘„*•¸uH:¬D°¿K  ÀÔÊ5™QD
             ÝF±E___€#³_#„)®Ü·àPq_)

_·XƒaBJÂ_ÓõÛ#gØ,+Š,Ñ‡I¨™k,p¼  ÿ‡N
Y_ì_
÷ZÕXcµÝ\![_p¢Ù¥_b__êÌoÔ_Ú((R  fL•»_r8…[¨WA
_,5>|EGë
Ý½ç5¹é  %¨AsgÃ†_KâN¥_är†_S_€a9436f3c9eff7e0d
8aab462822b50ed78f1619d8841c3ea5060d0fcf2bc83c
20c5cfd750e075118fccc1266ae945b49170272800332
33dba7bb13ebb948fd0a1E___€[øÛ[upDv•v¢ãª_  ©

¥_þÝ_9sÆ®£À__B¶Âöü  Ì™C³_tøù¸:_êÀKç:}˜¢“_ƒr
eZV___h_Ë¤ÝÇ… o_A8_#Ï†ÏcañFãž
          )h5]vlúW
²  œ‘„*•¸uH:¬D°¿K  ÀÔÊ5™QDÝF±E___€a“Yàa(L
8

Y_ì_
÷ZÕXcµÝ\![_p¢Ù¥_b__êÌoÔ_Ú((R  fL•»_r8…[¨WA
_,5>|EGë
Ý½ç5¹é  %¨AsgÃ†_KâN¥_är†_S_€a9436f3c9eff7e0d
8aab462822b50ed78f1619d8841c3ea5060d0fcf2bc83c
20c5cfd750e075118fccc1266ae945b49170272800332
33dba7bb13ebb948fd0a1E___€[øÛ[upDv•v¢ãª_  ©

¥_þÝ_9sÆ®£À__B¶Âöü  Ì™C³_tøù¸:_êÀKç:}˜¢“_ƒr
eZV___h_Ë¤ÝÇ… o_A8_#Ï†ÏcañFãž
          )h5]vlúW ²  œ‘„*•¸uH:¬D°¿K  ÀÔÊ5™QD
ÝF±E___€a“Yàa(L8
```

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: Masterkey Generation In The Encryption Process

```
E____5ZÛÇSñ ‡<Ç.#Þ-¼ƒbÎE_E___€_

¯šÁN¯Ä¹  7ƒgÙýÌñõ5Î_…__–  ”Y;@

µ«9Ub_‡kp®5  šÚx6m¨b

x_Ü†%ù´úcTò.:$4‘sÈ+‡ß•¶\²¹’²Àv  ^C

e5üÿÂ¦4__ Q
Š1Ù_•‰@ˆá€‘%Ãý}`Ò_Zg_Ÿñ_ÅÉºèž§ªE____GA_±]ß
y«Ø‹9x_…´\Ïá_E___€_€iÌCj8…ÙôçßÎÄú}_1°¦˜ë_z;ü_
¥½x_O™"Øõ–Åˆ  ÒÓ_NåUŠ

jÄw       $ktP_OOŽ»nþx
          Ýç`ûñ_Ÿ"“\úõ_Ú'éI’¡«wäO’_³C›÷ú3Œ_‹ÅP_¤i
X_J1úåÿ4?!†/_<dI7Õ

__E____{9-–†&iû/Z;Üw$[;È¬Á E___€

X_†Vv6+g
I_N]ä_<Ff¹⁄₂Õ_~û?  mÄVVO#™¬¸ëQBuíÝNËà&¢Óà_
ÒGEPøIé_Ó›×«ìä›–Ô#Ó,Å#_!<Nâ„)Ï·µ„0üà_†lË”ŽF

ÇäyDÿ'Û_me™_\ïŸ_KÜ_˜Gë¦_

UŸîÑ_Ð  ¥BE____>Asn«re_‘:®÷BÑë_ÄgîE___€Uqâƒ'
_
6Uú_JÈñ´oy_Ðf‹`7ó]Ñï#=ŠJdG$Á£¬¤~eIç_÷Ô)à=‰˜_È
ØnS^©"z—Â©0Ÿ;eprW©_K_¡¹€}_«  (W›o  Ã
·hßþbv³„FÌ_±        ÛéI_ß˜_Kò_Ñ_ûÂ˜™¤nÕ ˜4È—Ò
          ëã=Ë_÷à.ýe_ú
¡ðÄ^œ¤_ÿ_òYùH³:  <ÌŸ_|!ö</<â¢ñþ[uÜg

SØ©ÙÔü¾E____$_Áñ°  _{

_
¨›½.9OÛ©ƒ”E___€zTÌéW¯¡(}ß&  _Éy4_c£  u(¶4B,²ÜJ
ÀyòÙ˜›–
‡~öôßÞ_Í±¡Åºçð{s_!ˆK_æ_WezI_JYBu_'_“"4¶_Í ìRõlöc
¡ºÖ„+oN¼22Mh“¦_ Ëk9_ËªŠ_‹*‹Å»

Î£5¼  ƒP  9¢_è_ØE____
              ë…4ªa__fõöê3#ì_b_“E___€Dó‹*Ñ<]N!  À|l
            Šòúk=LÞÞ½_oC«+ˆýÃ  _óCÉÅ5ûoæ•Uß’G9ä
×@_€$ÎÅ¬V«

ÓCõ‰)F  _f”‘æšU"|  "±ëËž|©î___ªY[»þ  ø¨šy_2io¤Ò³
Þ
```

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: Public Key

```
E___€-Òv÷žJ‹_ÈÃ

Æ_ðv¹±Žûáôûo_Î®òa,>Ø‡çnãSÄ»I5Š‹^DcLÆúY
‰LÕ  _p÷$wOkí¼.3YP¶=‰ñõM˜²={Íá1±7ªä°:ç]
“:EÈÔ±[

…„_Ðx–
b¹UÛ  »_Ú¦__7Œ¾öE_éxÐ_¿.F¶>ùE___€6îm  ó
__¨³šŽÌ«Š«ÆQZ³)_û__!_F|  [ì_æ_õ_Ó»Õ3€~ÙÊ
û©ÂdÈ r‘˜öÙ€Ç’f¶

_è_Y0V‹skW÷¥Û3ÙŸ¹h2š!q¦·,…Ó²UõÁt’ "{J%n
\®—
¡øF#üä8[å__N'E_‰éÜ  ß_%jE___€_šõ‘¤A+?_O
_ô_“     M2

“b¬…_O
É_î,Ýˆ ôQ[=ôõ5š¨_(_t!ÄŒíý‹&_ôà›wŒèLïõ_åœ^ì
Z_±¾0®þãúÏ%ùîD"_yôÍ^î_Ì·êƒ_!çÿ;‡–
ŠõÝ!ÜÜî“?Õè”a_' Øœõ3Æ2
ˆÝÛQàVE___€O”:_âµ»Nøš  ï9þ  _ªå_n
…„µŒb¨  _Œù=  2†ƒôr
É#¦¾`Æ  åI QÓº#O¥8

&_£ß”  $½àê7é]ºùÎùê_îµ!_«+Qy«-
  ¦ã[zº\u¦ ÷_³*ß¤\·_÷æ$jBº_y

€Bëé_vÿžŸØÐ2E___€/_qIJ_%äÎòÖà…ÚÌÓ$›æÅ
¥? __Œ
¢_;†oÁ4xYÁgbµØû£
_¢_°_M+_RàãOñÆ\á„˜__ž_®„›ï`}ö`rÜèÄÃsFÍ[ö
Ý%¸˜šÂa1ƒÄÔ™m¡àÄ  _%qG=  ÿÑ:ðÞÃÄÓÙ‘
ÈÀSÛ^~tÎR_ E___€

v/íÅ“ˆ_ª
Ï  õ\_Î¢cçÜ__‹Ù6’À‹•ûRF5ã°ÊT_˜(à…Ç‰=Ê§ˆù
jE7ßp_,
ØH¹3ê‹¸á>†¶Ð

à  fm‹tövâÜŽõA¹
  Œ]tÂŠA>8Ä‡ÏÍÀÏ‘†QD_$À_öOsý¡c838D²º&¨ì
Þë_E___€  :žT…ÿ_Ò_&Áj›f\Î_Ú¸_àb·qfø¯÷j5_ù
Cv¬    »'6  à¨__EBàHd<  fm‹tövâÜŽõA¹
  Œ]tÂŠA>8Ä‡ÏÍÀÏ‘†QD_$À_öOsý  fm‹tövâÜŽ
õA¹  Œ]tÂŠA>8Ä‡ÏÍÀÏ‘†QD_$À_öOsý
```

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: Sample Encrypted Data For The Multi-User Data In Block Hash Framework

```
k™ˆ3bdÓ‹áŽ¬>[¨eO   Ë´   Ü†¬5óìão/ê_ë]¦"ü9-
´¬r‡7Ý‰mö«N:¯…Kâ(ù×¶¿*Ä{_8ìlSYÍBóó5$‹
           èqO‰‹+Pü›Õ
“HÉIP“FÑÕ+•   ÷‘rv¾–>æ
           :»£?>Sƒô3ÉN§pÁÚ“ÇÇÙ^[N   Rü·ÒË©íj¾vàºÞ
ÝÑjxil…c7$¡iZ<ú†¬›w×4Ë’˜‰qéOÌóÐyx¼>{Þ´NiÀ¸þÑÚ
ù|J¥Áú[˜   å“ÕÇø\/TcpûA¹
q‡…IÙÚ   Ìd}ÿ›®·ÍÕPªÿ
Bó

>ÏäÔ’[sJ£,ÏšGYŒR1ù1-G!|¡?ÈQÙÁ$Mü„òmÆƒàòš6ëØKóÎ
ôÿ/Ôä¡võš6ëØKóÎôÿ/Ôä¡võš6ëØKóÎôÿ/Ôä¡võš6ëØKóÎôÿ/Ô
ä¡võš6ëØKóÎôÿ/Ôä¡võš6ëØKóÎôÿ/Ôä¡võš6ëØKóÎôÿ/Ôä¡võ
š6ëØKóÎôÿ/Ôä¡võš6ëØKóÎôÿ/Ôä¡võš6ëØKóÎôÿ/Ôä¡võš6ëØ
KóÎôÿ/Ôä¡võš6ëØKóÎôÿ/Ôä¡võš6ëØKóÎôÿ/Ôä¡võš6ëØKóÎô
ÿ/Ôä¡võš6ëØKóÎôÿ/Ôä¡võzc
•D"®ç8-!Z.È<   ´žò   ÃañœkøÐ   wÑþVjNˆ=Õ”¹-
…g„â[_þvI³ºƒê’-..   ó5gâ˜É¦5\cÃ‡÷±ìVøÏYKEG’ó1ìã"F'D"#
ÍM$ä<êµ'Þ|,™¦X   tâq¾=   í´FWÿÊùcŠ³¢¬9i"   ž={tµ}³º
¾†w~`iz«hrÎ_ˆÁ˜‹O™Ÿ¡ƒÄO¬B[»%c•‘©.D$
```

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

r = (sum of all bytes c[j]) = 274

r = RMIN + (r mod S) = 6

0xe4c70355

c[j] = p[j] XOR c[j-1] XOR y = 0x0dec2ba8

r = (sum of all bytes c[j]) = 460

r = RMIN + (r mod S) = 2

0x1dc2956e

c[j] = p[j] XOR c[j-1] XOR y = 0x102ebec6

r = (sum of all bytes c[j]) = 450

r = RMIN + (r mod S) = 2

0xc1b63100

c[j] = p[j] XOR c[j-1] XOR y = 0xd1988fc6

r = (sum of all bytes c[j]) = 702

r = RMIN + (r mod S) = 4

0x2cdc890b

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

0xd58e349a

c[j] = p[j] XOR c[j-1] XOR y = 0x4153d262

r = (sum of all bytes c[j]) = 456

r = RMIN + (r mod S) = 3

0x76d51500

c[j] = p[j] XOR c[j-1] XOR y = 0x05bff74e

r = (sum of all bytes c[j]) = 521

r = RMIN + (r mod S) = 3

0x938a0782

c[j] = p[j] XOR c[j-1] XOR y = 0x9635f0c4

r = (sum of all bytes c[j]) = 639

r = RMIN + (r mod S) = 6

0x3f204758

c[j] = p[j] XOR c[j-1] XOR y = 0x46f92ebd

r = (sum of all bytes c[j]) = 554

r = RMIN + (r mod S) = 6

Final hash integers

[Ljava.lang.String;@3d075dc0    :=    Integrity    value
:40da75997f7489d1892228a577ff7089dd629071862d46
cee868ae68d5944e1fe935c49a02617322be47621857f50
99d257e66b7b3752d76c00b70cbef114cd2e33abde0959
b2d8e8b005aa39c3376c8916d266c724e16a10d73ffd0c0
5823e475935a33c2d98fb9aef875b55dc9c2c066debaaa7
c2877f

[Ljava.lang.String;@3d075dc0    :=    Integrity    value
:3d9a349e78ed518d8f48a7cb517cf5a7eacbab4d384c3d7
b23b9261311d7180c2c7a7bb096a64f02cd90d301ce7b2
e22e2fbc8aefbcf191eea8fd48a10a250
```

Table 5, illustrates the sample integrity value of the hash framework on a single hash record.

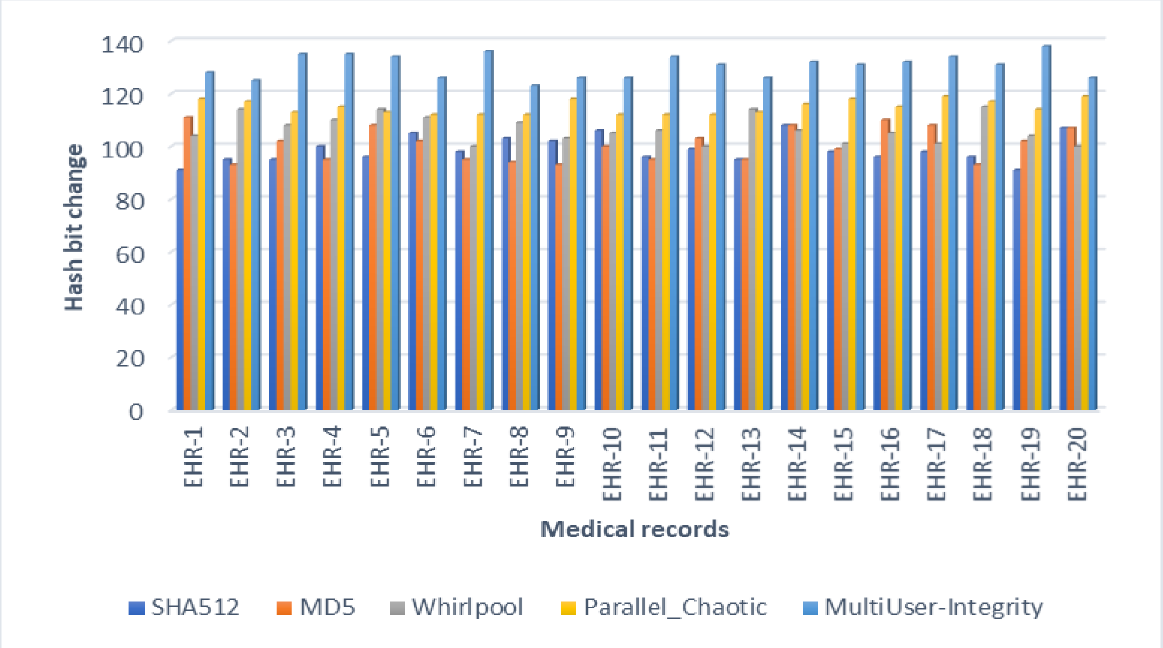

Figure 2: Performance Of Proposed Integrity Verification Model To Existing Hash Based Approaches On Variable Size Attributes (Hash Size =2048)

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, 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

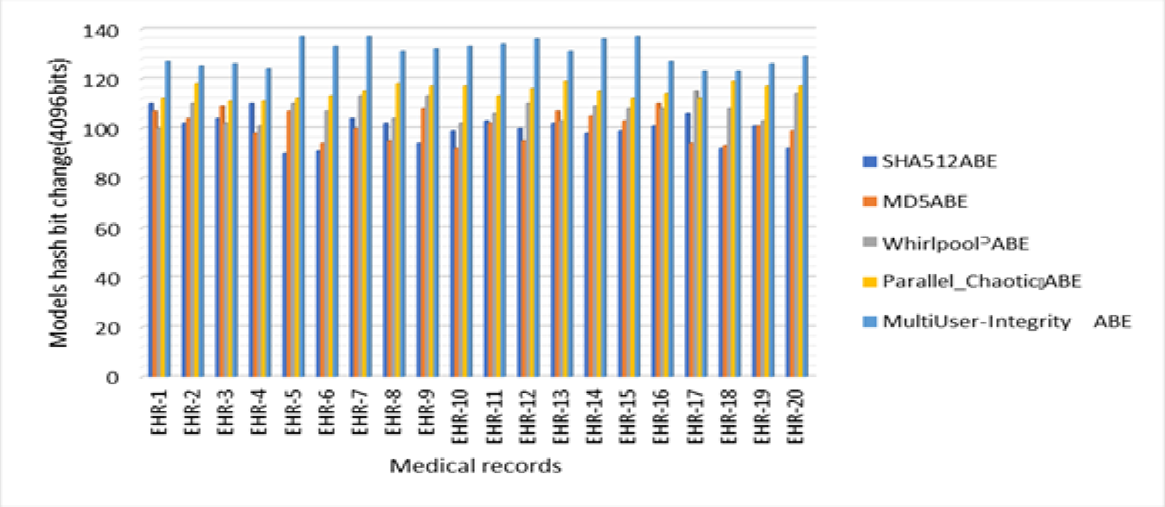

Figure 3: Performance Of Proposed Integrity Based Encryption Model To Existing Hash Approaches On Variable Size Attributes And Multi-User Data (Hash Size =4096)

Table 6: Performance Analysis Of Proposed Integrity Model And Conventional Approaches On Multi-User Data (Hash Size=2048)

Transactions	SHA512	MD5	Whirlpool	Parallel_Chaotic	MultiUser-Integrity
HR-1	5786	4846	5535	5355	4639
HR-2	4865	5620	5761	5452	4278
HR-3	5643	5027	5568	5566	4424
HR-4	5152	4988	4931	5837	4418
HR-5	5347	5402	5334	5383	4610
HR-6	5750	4922	5467	5777	4610
HR-7	5533	5324	4849	5452	4361
HR-8	5604	5723	5047	5370	4635
HR-9	5326	4881	5342	5387	4518
HR-10	4925	5592	5597	5150	4318
HR-11	5802	5620	4967	4870	4114
HR-12	5516	5242	5830	5412	4115
HR-13	4866	5108	5580	4940	4355
HR-14	4966	5385	5110	4949	4472
HR-15	5355	5025	5166	4982	4051
HR-16	4861	5028	4907	5308	4400
HR-17	5508	5098	5212	5320	4435
HR-18	5680	5403	5361	4920	4545
HR-19	5483	4902	5847	5790	4518
HR-20	5511	5630	5461	5797	4648

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)

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

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.

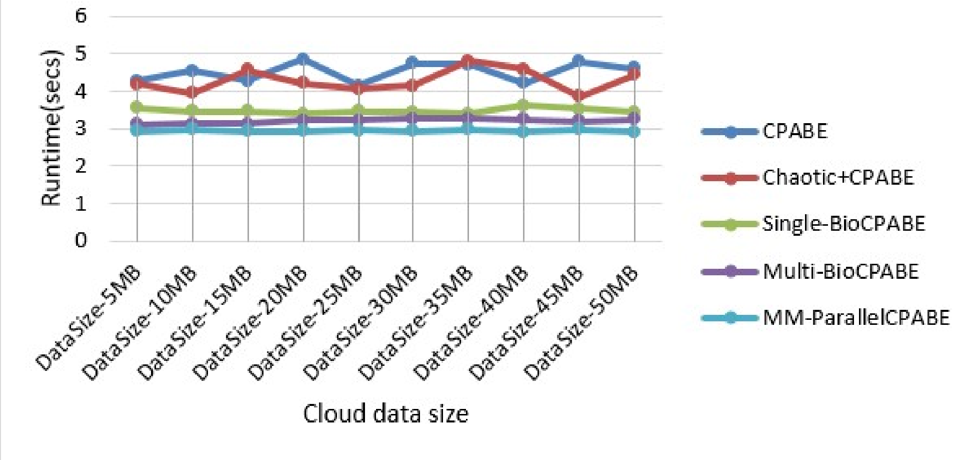

Figure 4: Performance Of Proposed Parallel Multi-User Based Encryption Model To The Traditional Integrity Based CP-ABE Models For Encryption And Decryption Runtime.

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)

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

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:

[1] F. Khoda Parast, C. Sindhav, S. Nikam, H. Izadi Yekta, K. B. Kent, and S. Hakak, “Cloud computing security: A survey of service-based models,” Computers & Security, vol. 114, p. 102580, Mar. 2022, doi: 10.1016/j.cose.2021.102580.

[2] H. Kaur and A. Anand, “Review and analysis of secure energy efficient resource optimization approaches for virtual machine migration in cloud computing,” Measurement: Sensors, vol. 24, p. 100504, Dec. 2022, doi: 10.1016/j.measen.2022.100504.

[3] S. Iqbal et al., “On cloud security attacks: A taxonomy and intrusion detection and prevention as a service,” Journal of Network and Computer Applications, vol. 74, pp. 98–120, Oct. 2016, doi: 10.1016/j.jnca.2016.08.016.

[4] R. Imam et al., “A systematic literature review of attribute based encryption in health services,” Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 9, pp. 6743–6774, Oct. 2022, doi: 10.1016/j.jksuci.2022.06.018.

[5] M. Sangeetha, P. Vijayakarthik, S. Dhanasekaran, and B. S. Murugan, “Fine grained access control using H-KCABE in cloud storage,” Materials Today: Proceedings, vol. 37, pp. 2735–2737, Jan. 2021, doi: 10.1016/j.matpr.2020.08.542.

[6] S. Ahmad, S. Mehfuz, and J. Beg, “Cloud security framework and key management services collectively for implementing DLP and IRM,” Materials Today: Proceedings, vol. 62, pp. 4828–4836, Jan. 2022, doi: 10.1016/j.matpr.2022.03.420.

[7] M. Mayuranathan, S. K. Saravanan, B. Muthusenthil, and A. Samydurai, “An efficient optimal security system for intrusion detection in cloud computing environment using hybrid deep learning technique,” Advances in Engineering Software, vol. 173, p. 103236, Nov. 2022, doi: 10.1016/j.advengsoft.2022.103236.

[8] O. Isaac Abiodun, M. Alawida, A. Esther Omolara, and A. Alabdulatif, “Data provenance for cloud forensic investigations, security, challenges, solutions and future perspectives: A survey,” Journal of King Saud University - Computer and Information Sciences, Oct. 2022, doi: 10.1016/j.jksuci.2022.10.018.

[9] W. Song et al., “Public integrity verification for data sharing in cloud with asynchronous revocation,” Digital Communications and Networks, vol. 8, no. 1, pp. 33–43, Feb. 2022, doi: 10.1016/j.dcan.2021.02.002.

[10] J. Tian, H. Wang, and M. Wang, “Data integrity auditing for secure cloud storage using user behavior prediction,” Computers & Security, vol. 105, p. 102245, Jun. 2021, doi: 10.1016/j.cose.2021.102245.

[11] B. D. Deebak and F. AL-Turjman, “Lightweight authentication for IoT/Cloud-based forensics in intelligent data computing,” Future Generation Computer Systems, vol. 116, pp. 406–425, Mar. 2021, doi: 10.1016/j.future.2020.11.010.

[12] M. Xu, D. Wang, Q. Wang, and Q. Jia, “Understanding security failures of anonymous authentication schemes for cloud environments,” Journal of Systems Architecture, vol. 118, p. 102206, Sep. 2021, doi: 10.1016/j.sysarc.2021.102206.

[13] M. Muhil, U. H. Krishna, R. K. Kumar, and E. A. M. Anita, “Securing Multi-cloud Using Secret Sharing Algorithm,” Procedia Computer Science, vol. 50, pp. 421–426, Jan. 2015, doi: 10.1016/j.procs.2015.04.011.

[14] R. Hayward and C.-C. Chiang, “Parallelizing fully homomorphic encryption for a cloud environment,” Journal of Applied Research and Technology, vol. 13, no. 2, pp. 245–252, Apr. 2015, doi: 10.1016/j.jart.2015.06.004.

[15] M. M. Potey, C. A. Dhote, and D. H. Sharma, “Homomorphic Encryption for Security of Cloud Data,” Procedia Computer Science, vol. 79, pp. 175–181, Jan. 2016, doi: 10.1016/j.procs.2016.03.023.

[16] B. Chen and X. Zheng, “Implementing Linear Regression with Homomorphic Encryption,” Procedia Computer Science, vol. 202, pp. 324–329, Jan. 2022, doi: 10.1016/j.procs.2022.04.044.

[17] H. Liu, Y. Xu, and C. Ma, “Chaos-based image hybrid encryption algorithm using key stretching and hash feedback,” Optik, vol. 216, p. 164925, Aug. 2020, doi: 10.1016/j.ijleo.2020.164925.

[18] C. E. J. Singh and C. A. Sunitha, “Chaotic and Paillier secure image data sharing based on blockchain and cloud security,” Expert Systems with Applications, vol. 198, p. 116874, Jul. 2022, doi: 10.1016/j.eswa.2022.116874.

[19] A. V. Tutueva, A. I. Karimov, L. Moysis, C. Volos, and D. N. Butusov, “Construction of one-way hash functions with increased key space using adaptive chaotic maps,” Chaos, Solitons & Fractals, vol. 141, p. 110344, Dec. 2020, doi: 10.1016/j.chaos.2020.110344.

[20] K. Ambika and M. Balasingh Moses, “An efficient SG-DACM framework for data integrity with user revocation in role based multiuser cloud environment,” Computer Communications, vol. 155, pp. 84–92, Apr. 2020, doi: 10.1016/j.comcom.2020.03.006.

[21] P. S. Challagidad and M. N. Birje, “Efficient Multi-authority Access Control using Attribute-based Encryption in Cloud Storage,” Procedia Computer Science, vol. 167, pp. 840–849, Jan. 2020, doi: 10.1016/j.procs.2020.03.423.

[22] M. Kumar, “Post-quantum cryptography Algorithm’s standardization and performance analysis,” Array, vol. 15, p. 100242, Sep. 2022, doi: 10.1016/j.array.2022.100242.

[23] R. Li, X. A. Wang, H. Yang, K. Niu, D. Tang, and X. Yang, “Efficient certificateless public integrity auditing of cloud data with designated verifier for batch audit,” Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 10, Part A, pp. 8079–8089, Nov. 2022, doi: 10.1016/j.jksuci.2022.07.020.

[24] M. H. Murtaza, H. Tahir, S. Tahir, Z. A. Alizai, Q. Riaz, and M. Hussain, “A portable hardware security module and cryptographic key generator,” Journal of Information Security and Applications, vol. 70, p. 103332, Nov. 2022, doi: 10.1016/j.jisa.2022.103332.

[25] B. Umapathy and G. Kalpana, “A novel symmetric cryptographic method to design block complexity for data security,” Computers and Electrical Engineering, vol. 104, p. 108467, Dec. 2022, doi: 10.1016/j.compeleceng.2022.108467.

[26] M. Bouchaala, C. Ghazel, and L. A. Saidane, “TRAK-CPABE: A novel Traceable, Revocable and Accountable Ciphertext-Policy Attribute-Based Encryption scheme in cloud computing,” Journal of Information Security and Applications, vol. 61, p. 102914, Sep. 2021, doi: 10.1016/j.jisa.2021.102914.

[27] M. Swetha and M. Latha, “Security on mobile cloud computing using cipher text policy and attribute based encryption scheme,” Materials Today: Proceedings, Aug. 2021, doi: 10.1016/j.matpr.2021.06.462.