A Decentralized Accountability Framework for Enhancing Secure Data Sharing Through ICM in Cloud

Dhanapal.R, Tharageswari.K, Karthik.S

Abstract*— The Cloud substitutes a computing criterion where shared configurable resources are afforded as an on-demand service over the Internet. Moreover, the cloud environment provides resources to the users on the basis of services like SaaS, PaaS and IaaS. Generally, a cloud can be referred as private cloud or public cloud. When a Cloud Service Provider (CSP) imposes upon public cloud resources to compile their private cloud, the result is demonstrated as a virtual private cloud. Private or public, the imperious intent of cloud computing is to provide simplistic, reliable usage of various computing resources. One of the significant features of cloud is that the outsourced data are accessed through any anonymous machines over the Internet. On the other hand, it creates an issue that user's fear of unknown access of data, which can become a major difficulty to the wide implementation of cloud. In this paper, a decentralized accountability framework is developed to monitor the actual usage and access of the data that is shared on cloud. For that, a logging mechanism that includes authentication for each user to access the data has also been provided. Moreover, some procedures for providing the data under the control of data owner includes Integrity Checking Mechanism (ICM) have also been developed. The overall process strengthens the security constraints over cloud. And the experimental results reveal that the approach affords secure and scalable data sharing with reduced memory utilization and processing time.

Index Terms*— Cloud Computing, Accountability, Data sharing, Authentication, logging, ICM.

I. INTRODUCTION

In current scenario, Cloud computing provides novel way to increment the effective utilization for IT services oriented to the Internet services, by affording dynamic scalability and often virtualized computing resources as a service over the cloud. Furthermore, there are a number of notable individual and commercial cloud computing services. Cloud providers such as Amazon, Microsoft, Google, Yahoo and Sales force [12]gives several promising services to the users like storing data and accessing it from any location, file sharing and online services to support sales and business etc,.

An appropriate definition of cloud computing has been given by the National Institute of Standards and Technology (NIST). It denotes that cloud computing is a concept for referring adaptable and access to the requirement of users to a wide collection of composed computing resources such as storage, application, servers, networks and software which can be equipped and provided quickly with a promising interaction between the provider and the end user.

The features of cloud combine rapid elasticity, wide network access, on demand self service and resource pooling. Mainly the cloud services can be accessed via public, privte, community or hybrid cloud. Figure 1 depicts the various applications on cloud that are categorised as Platform as a Service (PaaS), Infrastructure as a Service (IaaS) and Software as a Service (SaaS). Among these, the focus of this paper is about Data Storage as a Service and method to overcome its security constraints.

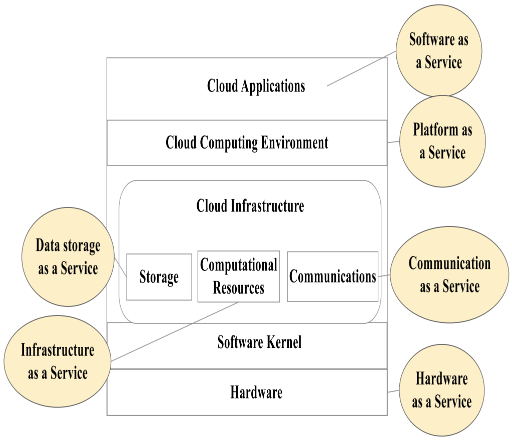

Figure 1: Various Cloud Services

With all those advantages of cloud, it is also acceptable that there are so many security issues. When there are some data outsourced through cloud, there are number of issues such as, accountability, integrity of data etc, to be considered [9]. Data security is an important issue in cloud computing and also it is very complicated to provide data integrity and confidentiality. To that concern, this paper covers the major security aspects such as Integrity, Confidentiality and Availability (ICA).

Manuscript published on 30 August 2019.
***Correspondence Author(s)**

Dhanapal.R, Assistant Professor, Department of Computer Science Engineering, Karpagam Academy of Higher Education, Coimbatore-641021, Tamilnadu, India. (E-mail: dhanapalramasamy.p@gmail.com)
Tharageswari.K, Assistant Professor, Department of Computer Science Engineering, Karpagam Academy of Higher Education, Coimbatore-641021, Tamilnadu, India. (E-mail: tharahari1515@gmail.com)
Karthik.S, Professor & Dean, Department of Computer Science and Engineering, SNS College of Technology, Coimbatore, Tamilnadu, India. (E-mail: profskarthik@gmail.com)

As it is shown in figure 2, integrity deals with protecting the data being modified by an unauthorized user, confidentiality concentrates on preventing the data from unauthorized access and finally availability provides the on demand access to the authorized person. On concentrating the above defined security aspects on cloud, this paper deals with secure authentication using Identity based Encryption technique. Generally, it is defined as a public key encryption mechanism uses an arbitrary string as public key that can be the unique data given by the user such as phone number, date of birth etc,. As in conventional encryption techniques, the IBE mechanism does not require public key infrastructure or certificate administration.

And considering the fear of anonymous data access over the stored data on cloud, it is very vital to provide an efficient mechanism to the data owners to track their data usage or accessibility over cloud. So it is significant to guarantee the access control of data. In order to track the proper data usage, log files are generated and monitored periodically. Traditional access control models are not appropriate for this decentralized environment since it has been developed for closed domains like operating systems and databases. Contrasting privacy protection technologies, data confidentiality covets on keeping the data access traceable and transparent. The proposed framework develops a point to point accountability in an extremely distributed manner. Moreover, the major inventive attributes of the proposed work lies in its capability of handling powerful and trivial accountability, which combines the aspects of control over the shared data and authentication. That is, the data owner can monitor the service level agreements, but also imposes the usage and access control rules as mentioned.

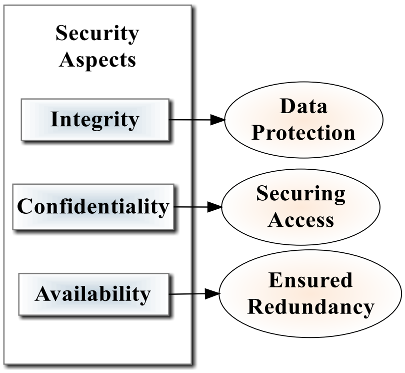

Figure 2: Cloud Storage Security Aspects

In addition to that, ICM (Integrity Checking Mechanism) has also been developed to improve the efficacy of the overall process. Further, the accountability framework affords abundant confrontations, comprising distinctively identifying CSPs, conforming to a highly decentralized model, establishing the reliability of the enrollment, etc. The basic intent is to enhance the programmable capability of JAR files (Java Archives) to routinely audit the usage of the user's data in accordance with cataloguing policies and access control policies that are to be incorporated within the JAR files to CSPs. Memory optimization has also been achieved through this, that has been a major area while discussing about cloud.

Any operations to the data will timely a reliable and automated logging mechanism. The Archive files are provided with a middle layer of contact that forms a secure channel. Additionally, a detailed security analysis has been launched with the compromised Java Running Environment (JRE). The remainder of this paper is framed as follows: Section 2 provides a description on the related work. Section 3 lays out the proposed DAF mechanism. Section 4 experimental analysis and comparison charts and Section 6 concludes the paper with paths to future enhancement.

II. RELATED WORKS

In this section, some of the works that are concentrating on the security and privacy problems on outsourcing the data over cloud are discussed. A model for provable data possession (PDP) was proposed in [14]. It is stated that remote data monitoring is processed by the model and it organizes huge data sets in decentralized storage network. Nevertheless, it permits the end user who has stored data in an untrusted cloud server to check whether the server provides promising service to the client without acquiring it. It significantly reduces the input/output costs.

In [1], the computational overhead problem was solved by the traditional key distribution mechanism. The paper combines three encryption algorithms: Attribute Based Encryption (ABE), Proxy re-encryption and lazy re-encryption. It was claimed that the accountability was achieved by the private key of data owner that tends to the complex processing. The ABE mechanism is incorporated in another process for securing the health data over cloud. The authors have used multi data owner criteria for minimizing the key management problems between data owner and the user. Following that, the authors of [2] described a mechanism for solving the information leakage by indexing in the cloud. Specified data binding technique was incorporated for that problem. It was explicitly stated that the security issues on cloud could be concentrated on further implementations.

Identity Based Encryption (IBE) enforces an substitute security standard to the traditional public key systems [11]. The core trait of IBE is that public keys are not generated based on the network, but they are adeptly evaluated from the user's distinctive identity data which is provided on the cloud. Some of the IBE related applications are listed below.

- Time oriented cryptography, in which data decryption can be done based on the sender specification
- Time moment the future
- Hierarchical IBE (HIBE)

Role-based access to secured data in the cloud data security was sensibly covered in [7]. The paper described a secure network architecture that has the Third Party Auditor (TPA). The major constraint there is about the reliability of the TPA. The further work in [3] had been processed with the bilinear aggregate signature to support multiple users at an instant along with the security issues in cloud. The accountability measures could be clearly defined.

When the outsourced content in the cloud is stored for more than a period of time, organizations and users unavoidably will have security issues. There may have the doubt that the data that is stored in the cloud is really safe or not and whether users in the future can get to that content or not. For solving the problem, a security and protective storage of evidence protocol has been developed in [10]. But, the logging mechanism was not effectively described.

Merkle Hash Tree (MHT) has been enforced in [13] for block authentication that effectively tightens the verification process for secure access of shared data. Asha has narrated a survey paper [4] about the various security and privacy concerns of cloud computing. Moreover, the paper narrated the places to be concentrated more, while providing security. The attributes discussed in that paper are as follows:

- Client side security
- Security issues from CSP
- Minimal access control over data
- Network security
- Data encryption
- Data revitalization
- Securing cloud content
- Installation and maintenance of firewall
- Certification and auditing
- Backup and recovery
- Identity and access management

Flexible distributed storage integrity and auditing mechanism has been proposed in [6]. The mechanism solves various attack. That is, the paper focused on integrity analysis, but the major limitation is about the accountability of the TPA. It was also claimed that the TPA should not request for any local copies of the shared data and include any vulnerability to the CSP or user's system. HASBE (Hybrid Attribute-set Based Encryption) was a different technique incorporated in [19]. This encryption is an enhancement of cipher text-policy attribute-set-based encryption (ASBE) with a multi-level organization of users. It was given that the scalability was effectively achieved with this technique.

The paper [20] proposed a pairing-based provable multi-copy data possession (PB-PMDP) scheme, which provides proof to the customers that all data that are stored over the cloud are secure and authorized. In [22], Cloud Information Accountability (CIA) model was derived and effectively focused on Information Accountability. The methodology provided usage of shared data, transparent and traceable. There is another review work [5] that helps to learn more concepts on cloud. The paper was focused on privacy and security concerns while storing data on cloud and methodologies for overcoming those issues. In [15], Subashini et al., have discussed about the security issues separately in IaaS, PaaS and SaaS. It has been a great work that defines some security models as well.

Pairing based provable multi copy data possession (PB-PMDP) is proposed in [21]. The methodology ensures the user that the outsourced data is safe and remains unchanged. Liu et al., have developed MONA [16] which uses multi owner data sharing and for privacy preserving dynamic broadcast encryption method is used. Another work called RS-IBE (Revocable-Storage Identity Based Encryption) in [17]. The authors primarily concentrated on removing the unauthorized user from the network. Moreover, the security goals framed on the basis of data confidentiality, forward secrecy and backward secrecy. But the concept is a bit complex during the real time implementation. Finally, Zissis et al,[18] narrated the various security issues and defined the security requirements to overcome those issues. Several efficient solutions have also been demonstrated in the paper for removing those security threats over cloud. In [23], real-time health monitoring model has been proposed using Ad-hoc networks and the authors of [24] and [25] discussed about the model for secure and scalable data sharing between nodes in the communication networks.

III. PROPOSED WORK

As is well known, cloud provides services over the internet. It stands that the resources are accessed from unknown hosts. Hence, there arise problems that data owner may have the anonymous access over the shared data on cloud. In order to solve that problem and to provide more security over the shared data on cloud, the proposed work focused on developing a Decentralized Accountability Framework (DAF) that comprises enhanced techniques for proper authentication, encryption and integrity. Identity Based Encryption is being processed for securing access and Log files are generated for tracking the usage of the outsourced on the cloud. Distributed auditing has been performed and an efficient ICM has also done for checking the data which remains intact. By this mechanism, the usage of shared data is being tracked and providing more security by blocking that unauthorized user from accessing the data.

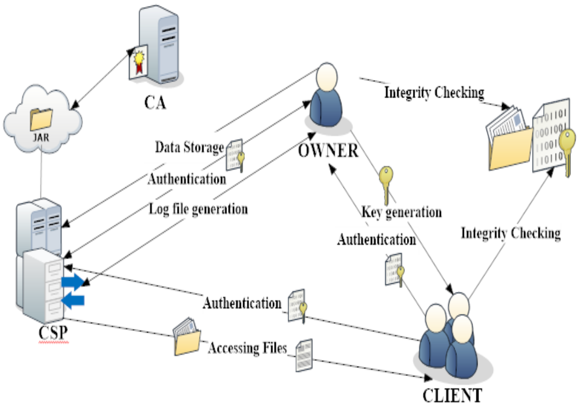

Figure 3: Overview of Decentralized Accountability Framework (DAF)

3.1 Decentralized Accountability Framework:

The Decentralized Accountability Model developed in this work conducts automated sorting with enhanced authentication by IBE which is followed by the JAR file creation of the particular data that is to be outsourced on the cloud. Log files are generated and distributed auditing of pertinent access accomplished by any element, processed at the respective cloud service provider at any instant.

The authentication process is efficiently performed by acquiring the unique specifications of the data owner, who are willing to outsource the data on the cloud and a unique key is generated for each user in the basis of Identity Based Encryption (IBE) mechanism. Then, ICM (Integrity Checking Mechanism) has been performed to check the originality of the outsourced one and the log file updates. After finding the malicious user, it is being blocked from the data access. In IBE, the key is generated in accordance with the unique information submitted by the user. Basically, this process is for tightening the authentication process, thereby enhancing the security on cloud.

The Figure 3 presented above shows the overview of the distributed accountability framework. It is explicit from the figure that logging of data owner with their information on to the cloud and logging of client on to the data owner's host are the significant tasks to be performed prior. Moreover, CA is the Certification Authority that shows that the corresponding cloud is more secure and reliable to process over the data on the network. This is for creating an impact about the reliability of the cloud in which the data owner shares his data.

As mentioned earlier, the key generation part has been done with the IBE mechanism. When the authentication is verified, the user is permitted to access the data combined in the archive file (the compressed format of data and logger that are stored on the cloud). Depending upon the configuration assignments described at the time of archive framing, it will afford access-control combined with logging. As for the logging, each instant there is an access to the outsourced content; the JAR will typically produce a log record, securing it using the public key and store it along with the data on CSP (Figure. 3). The encryption of the log data averts unauthorized access or modifications to the log records by adversaries. The data flow of the overall process is illustrated in figure 4. It is obvious from the figure that the auditing is performed between the CSP and the data-owner to get the information about the shared data and to block the unauthorized access. The auditing mechanism has two main advantages.

- It assures higher rate of log record accessability.
- It also minimizes the human work load.

The significant part is the enforcement of Integrity Checking Mechanism (ICM) for checking the modifications over the data.

3.2 Data Storage on Cloud

In this methodology, the data storage on cloud composed of two major things: logging has to be done with the CSP and compressing the data as JAR files. Here, IBE method is used for authentication and data compression has been performed for optimal memory usage of cloud.

Generally, IBE eliminates the use of Public Key Infrastructure. The data owners using IBE do not need any public keys and its respective certificate from the receiver, since it uses the combined identities such as emails or IP address paired with common public parameters for encryption. The process is done on the basis of the follows four steps: 1. Setup, 2. Extract, 3. Encryption and 4. Decryption

1. Setup: Assuming a Security constant *SP* and returns *para* (system attributes) and a Master_key. The system parameter combines depiction of a fixed message sector *MS* and a depiction of a fixed cipher text sector *CP*. It seems that the system metrics will be known openly, while the master_key will be precise only to the secret key producer.

2. Extract: Assuming IP para, master_key and an unique ID, which is $ID€\{0,1\}^*$ and returns a secret_key *a*. Here, ID is taken as an random string that will be utilized as the public_key for encryption and *a* is its respective decryption key. The process determines its secret key (secret_key) from the mentioned public key.

3. Encryption: with provided IP para, ID and MS€*ms*, it gives the cipher text as CT€*ct*.

4. Decryption: with the provided input CT€*ct* and the secret key *a*, it generates MS€*ms*

The process should assure the ordinary uniformity limitation, specifically when *a* is the secret_key which is produced by the extort process, when it is taken ID as the public_key, then,

$$\forall \mathrm{MS€ms}: \mathrm{Decrypt}(\mathrm{para}, 0, \mathrm{a}) = \mathrm{MSwhereCT} = \mathrm{Encrypt}(\mathrm{para}, \mathrm{ID}, \mathrm{MS}) \qquad (1)$$

Then, owner sends the archive to the cloud service provider that the owner assents to. In order to validate the CSP to the JAR, a trusted Certificate Authority (CA) has been used. Further, the JAR file implementation also affirms automatic log data operations, which meets the all the scalable requirements on data sharing over cloud.

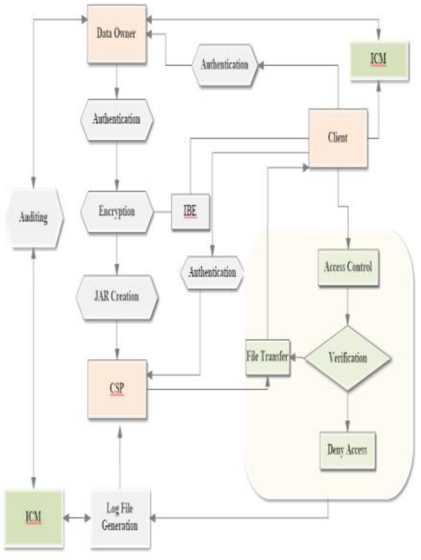

Figure 4: Data Flow of the Proposed Work

3.3 Log File Generation

The next process of the work flow is to create log files using a logger component to monitor the actual data usage on cloud. The log record is prolonged at the cloud and it will be updated regarding the changes made over the outsourced data. Here, the record is updated on the basis of changes on JAR and new log entities are appended in sequential order on to the cloud service provider.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

During Log file creation, it is stated that, Log File (*LF*)=(file$_1$,....,file$_n$). Here, each file$_i$ is encrypted distinctly and enrolled at the log file. Specifically, a log record that is appended on to the *LF* takes the following form (2),

$$file_i=\langle ID,AccP,T,Loctn,CS(ID,AccP,T,Loctn)|file_{i-1}|\ldots|file1),snD\rangle \quad (2)$$

Here, $file_i$ denotes that an entity specified by ID has processed an action AccP, which indicates the access privileges on the data outsourced at instant T at location Loctn. Futher, the component *CS (ID,AP,T,Loctn)|file$_{i-1}$|…|file$_1$)* represents the checksum of the log records leading the newly added data and then it is combined with the main data of the log file itself. The checksum is evaluated using a collision-free hash function. The component *snD* denotes the sign of data that are produced by the cloud server.

3.4 Distributed Auditing

As mentioned before, auditing is the mechanism, enforced here for tracking the actual usage of data. Push and Pull are the two auditing functions incorporated here. While in push condition, periodically the log record is automatically updated to the data owner, whereas the pull active is an approach works on requirements in which the record is available to the data owner only when there is a requirement. Pull action is initiated in two cases: when occurs a time elapse with respect to the predefined time settings and the when the stored JAR file surpasses its defined size. On the other side, the pull active enables when it receive the request from the auditor the pull message composes an FTP pull command and the user is loaded with the location of the data and also an enclosed duplicate of the preserved log record. In a case, when there are multiple loggers for the similar data items, the log organizer will combine log records from them before forwarding to the data owner. The log organizer is also liable for reliable file maintenance. Additionally, the log organizer can itself accomplish log creation in addition to the distributed auditing. Further, extrication of auditing functions and log creation enhances the cloud service.

3.5 Integrity Checking Mechanism (ICM)

Once the stored data of a data owner is accessed by a client, it should be checked under ICM. The integrity of data will be checked for the data consistency. Before data outsourcing, the data owner has to be intended generate the meta-data for integrity verification.

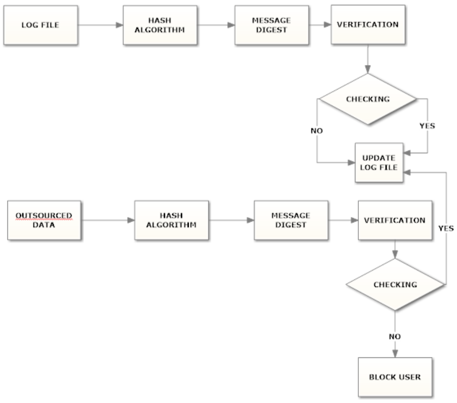

Figure 5: Block of ICM

The meta-data is produced using the cryptographic hash functions and Message Digest functions (Message digest). When there is a user gets appropriate access control with respective privileges (read/write), the user is allowed to do so. Nevertheless, the integrity checking is performed for checking whether the obtained data is modified by any untrusted or unauthorized user. In such cases, the mechanism would incorporate some blocking procedures to cancel the given access of that particular user and prevents from outsourcing the owner's content in modified manner. Furthermore, the log file has also been included in this operation for reporting to the data owner in two manners, named push active/ pull active during the distributed auditing process. The ICM process also invoked on the log file generation process to make the log files more secure i.e. to prevent modification on log file data. The process of IAE is further explained with the following block diagram (Figure 5). The following algorithm defines the working process of Hashing using the hash functions SHA-1 and MD5, with the assumption of bock size as 64 bytes.

Hash Algorithm for ICM:

```
Function string hashcode (Key (k), Content (c))
    If length (k)> sizeofblock) then
        Key=hash (k) // shortens the keys that are longer than block size
    End if
    If length (k)< sizeofblock) then
        K= k‖[0×00* (sizeofblock-length (k))] // the key k is zero padded when it is shorter than the size of block, where ‖denotes concatenation
    End if
    OP_key_pad=[0×5c*sizeofblock]XORkey    // where size of block is that of the elemantary hash function
    IP_key_pad=[0×36*sizeofblock]XOR key
    ReturnHash(OP_key_pad‖hash(IP_key_pad‖content))
    String 32 VAL= MD5(HashCode(Key, Content))
End function
```

The integrity checking will be performed in two places: on the log files that are transmitted between the CSP and the data owner and on the shared data. As per the aforementioned conditions, the access for the user will be blocked, if there is any insecure activities are found.

IV. EXPERIMENTAL RESULTS

In the experimental analysis, the proposed methodology has been analyzed with respects to the factors such as time, memory utilization, efficiency and the computational overhead. Furthermore, it has also been compared with the previous methods such as HASBE, HIBE and CIA. Comparison is done for memory utilization and time efficiency respectively. With respect to particular instant, the overhead can take place at three scenarios: during the validation process, during log file encryption, and while the assimilation of the logs.

Further, the archive files act as a compressor of the logs that it maintains. The first process is occupied in time evaluation to produce a log file when there are some entries continuously using the data that causes constant logging. The chart in figure 6 depicts that the time taken for log file generation.

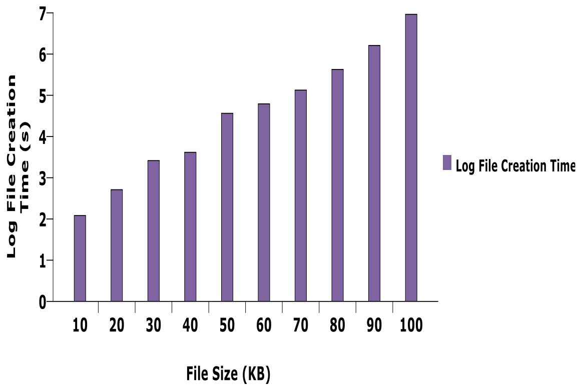

Figure 6: Time Taken for Log File Creation

In order to verify whether the process is with bottleneck, the amount of time needed for combining the log files has been evaluated. In this analysis, it is validated that each of the log files had 11% to 26% percent of the files in similar with one another. The appropriate number of files that are common is arbitrary for each recurrence of the examination. The time has taken as the mean over 15 iterations. The time has been analyzed to merge up to 60 log files of 100 KB, 200 KB, 300 KB, 400 KB and 500 KB each. The results are displayed in Figure 7. The comparison of the result between the aforementioned methods is given in Figure 8.

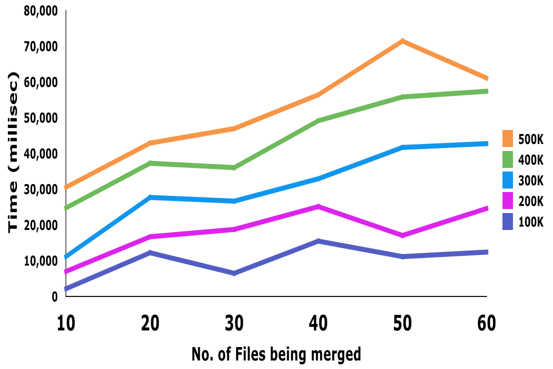

Figure 7: Time Taken for Log Merging

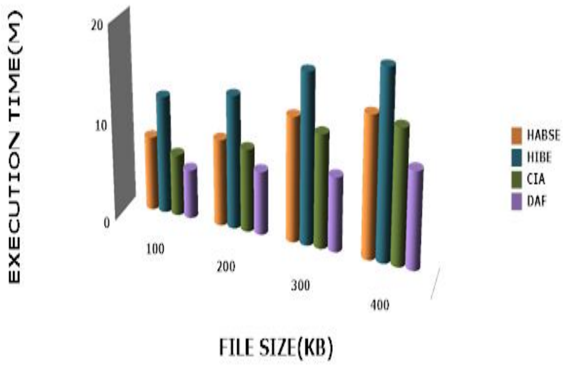

Figure 8: Overall Execution Time

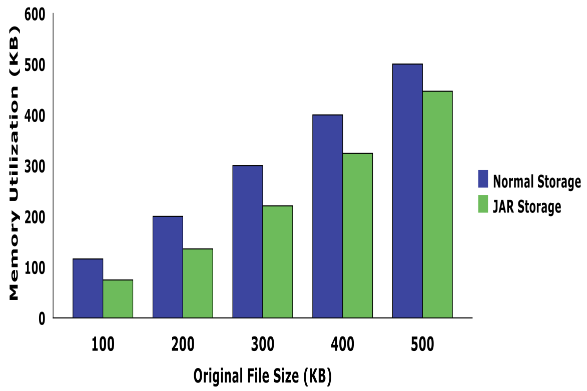

Figure 9: Analysis on Storage Overhead

Furthermore, the analysis has been made on storage overhead. The size of the archives is measured by changing the number data sets and size of records held by them. The storage of data on cloud is compared here as normal file storage and JAR file storage. While analyzing, it is noted that the JAR storages acquires less memory. Hence, the proposed work provides better memory utilization (Figure. 9) and its comparison with other models is given in Figure 10.

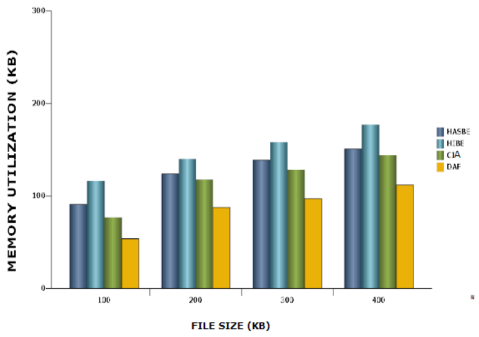

Figure 10: Comparison on Memory Utilization

V. CONCLUSIONS AND FUTURE WORK

Cloud computing has established a variety of significant privacy and security concerns, due to the verity in the cloud about shared data and applications endure on the cloud paradigm, which has been effectively managed by a third party. The problem is identified effectively by the literature survey. Hence, a efficient highly Decentralized Accountability Framework to monitor the appropriate storage of the user's data continuously in the cloud. By the incorporation of the log record model using IBE the data will be accessed by the authorization control provided by the data owner. Distributed auditing provides the way to track data usage and by integrity checking mechanism a secure data sharing is detained in the cloud, hence the data owner need not worry about the data modifications. The mechanism also involves in preventing the data sharing from security attacks such as copying attack, disassembling attack and man-in-the-middle attacks.

Moreover, the process also reduces the computational overhead and storage overhead on the cloud storage servers. The previous section shows the implementation work adequately. In future enhancement, hybrid cryptographic techniques can be used for key generation and the implementation could be checked for further security issues in cloud by using various advanced cryptographic functionalities.

REFERENCES

1. Shucheng Y, Cong W, Kui R and Wenjing L, "Achieving Secure and scalable, Fine-grained Data Access Control in Cloud Computing," In the Proceedings of IEEE Conference-INFOCOM, March 2010, pp. 1-9.
2. Squicciarini A, Sundareswaran S and Dan L, "Preventing Information Leakage from Indexing in the Cloud," In the Proceedings of 2010 IEEE Transactions on Distributed Computing, July, 2010, pp. 188 - 195.
3. Cong W, Qian W, Kui R nad Wenjing L, "Privacy-preserving Public Auditing for Data Storage Security in Cloud computing," IEEE Transactions on Parallel and Distributed Systems, San Diego, March 2010. ISBN: 1063-6692.
4. Asha M, "Security and privacy Issues of Cloud Computing: Solutions and Secure Framework," International Journal of Multidisciplinary Research, April, 2012, Volume: 2, Issue: 4, pp. 182-193.
5. Selvamani K, Jayanthi S, "A Review on Cloud Data Security and Its Mitigation Techniques", Science Direct-Elsevier, International Conference on Intelligent Computing, Communication & Convergence (ICCC-2015) science direct. Procedia Computer Science, Vol. 48, 2015 pp. 347 – 352.
6. Cong W, Qian W, Kui R, "Towards secure and dependable storage services in cloud computing," IEEE Transactions on services computing, 2012, Volume: 2, Issue: 2.
7. Cong W and Kui R, Jin L, Wenjing L, "Toward publicly auditable secure cloud data storage services," IEEE Transactions on Secure Computing, July /Aug-2010 ISBN: 0890-8044/10.
8. Giuseppe A, Randal B, Reza C, Joseph H, Lea K, Zachary P and Dawn S, "Provable data possession at untrusted stores," In the Proceedings of 14th ACM conference on Computer and communications security (CCS '07), 2007, pp. 598-609.
9. S. Pearson and A. Charlesworth, "Accountability as a Way Forward for Privacy Protection in the Cloud," Proceedings First International Conference Cloud Computing, 2009.
10. Narn L and Yun-Kuan C , "Hybrid Provable Data Possession at Untrusted Stores in Cloud Computing," In the Proceedings of IEEE 17th International Conference on Parallel and Distributed Systems (ICPADS), December, 2011, pp. 638 - 645.
11. Kihidis, A. , Chalkias, K. and Stephanides, G. , "Practical Implementation of Identity Based Encryption for Secure E-mail Communication," In the Proceedings of 14th Panhellenic Conference on Informatics (PCI), September, 2010, pp. 101 – 106.
12. P.T. Jaeger, J. Lin, and J.M. Grimes, "Cloud Computing and Information Policy: Computing in a Policy Cloud?," J. Information Technology and Politics, vol. 5, no. 3, pp. 269-283, 2009.
13. Qian W, Cong W, Kui R and Wenjing L, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing," In the Proceedings of IEEE Transactions on Parallel and Distributed Systems, May 2011, Volume: 22, Issue: 5, pp. 847-859.
14. Giuseppe A, Randal B, Reza C, Joseph H, Lea K, Zachary P and Dawn S, "Provable data possession at untrusted stores," In the Proceedings of 14th ACM conference on Computer and communications security (CCS '07), 2007, pp. Pages 598-609.
15. S. Subashini, V.Kavitha,"A survey on security issues in service delivery models of cloud computing" Journal of Network and Computer Applications, Vol- 34,2011, pp. 1–11.
16. Xuefeng L, Yuqing Z, Boyang W and Jingbo Y, "Mona: Secure Multi-Owner Data Sharing for Dynamic Groups in the Cloud, " IEEE Transactions On Parallel And Distributed Systems, Vol. 24, No. 6, June 2013 Pp.1182-1191.
17. Jianghong W, Wenfen L, Xuexian H, "Secure Data Sharing in Cloud Computing Using Revocable-Storage Identity-Based Encryption, " IEEE transaction on cloud computing, Journal Of Latex Class Files, VOL. 14, NO. 8, August 2015, pp 1-13.
18. Dimitrios Z, Dimitrios L, " Addressing cloud computing security issues" science direct-Elsevier, Future Generation Computer Systems, Vol. 28, 2012, pp. 583–592.
19. Zhigou W, June L, Deng, R.H., "HASBE: A Hierarchical Attribute based Solution for flexible and Scalable access control in cloud computing," IEEE transactions On Information and Forensics and Security, April-2012, Volume: 7, Issue: 2, pp. 743 – 754.
20. Ayad F. Barsoum and M. Anwar Hasan, "Integrity Verification of Multiple Data Copies over Untrusted Cloud Servers," In the Proceedings of 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID 2012), pp. 829-834.
21. Ayad F. Barsoum, M. Anwar Hasan, "Integrity Verification of Multiple Data Copies over Untrusted Cloud Servers" 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, 2012, pp 829-834
22. SmithaSundareswaran, Anna C. Squicciarini and Dan L, "Ensuring Distributed Accountability for Data sharing in the Cloud," IEEE Transactions of Dependable and Secure Computing, August 2012, Volume: 8, Issue: 4, pp. 556- 568.
23. Dhanapal, R & Visalakshi, P 2016, 'Real Time Health Care Monitoring System for Driver Community Using Adhoc Sensor Network', Journal of Medical Imaging and Health Informatics, vol. 6, no. 3, pp. 811-815.
24. Dhanapal, R & Visalakshi, P 2016, 'Optimizing Trust Based Secure Routing for Unified Efficient Resource Sharing for Large Scale MANET-TSRRS', Asian Journal of Information Technology, vol. 15, no. 19, pp. 3756-3762.
25. Dhanapal, R & Visalakshi, P 2018, 'An Efficient Model for Secure and Scalable Health Log Mangement in Cloud using EH-ABE', International Journal of Engineering Technology, vol. 7, no.4.19, pp. 309-318.

Dhanapal.R was born in India. He received the B.Tech and M.E. degrees in Information Technology and Computer Science engineering from Anna University, Chennai, Tamilnadu, in 2007 and 2011, respectively, and the Ph.D. degree from the Anna University, Tamilnadu in Information and Communication Engineering in 2018. He was a faculty member in the Department of Computer science from 2012. Currently works in Karpagam Academy of Higher Education as an Assistant Professor.