An Efficient & Secured Way of Sharing the Data Using Distributed Accountability & Auditing in Cloud

Chandan.M, Aparna K.S

Abstract— **Cloud computing can be defined as the set of hardware, networks, storage, services, and interfaces that combine to deliver aspects of computing as a service. In cloud data storage system, users store their data in the cloud and no longer possess the data locally. User's data are usually processed remotely in unknown machines that users do not own or operate. The lack of confidence in trusting information flow in cloud has become common, as users fears of losing control of their own data. We leverage the JAR programmable capabilities to both create a dynamic and traveling object, and to ensure that any access to users' data will trigger authentication and automated logging local to the JARs. To strengthen user's control, we also provide distributed auditing mechanisms. In addition to this some privacy protection techniques only concentrate on preventive controls; research is needed for detective controls in the area of accountability and auditing. This paper presents a framework for distributed accountability and auditing which is used to protect user's data and also tracking the actual usage of data in the cloud. In particular, a logging mechanism is provided for the user's data with access policies, and ensures that any access to their data will trigger authentication, by this mechanism data owner may know his/her data is handled as per his access policies.**

Index Terms-- **Accountability framework, auditing, cloud computing, data sharing, logging, security.**

I.INTRODUCTION

Cloud computing is a collection of IT resources which are deployed in a network. By deploying IT infrastructure and services over the network, an organization can purchase these resources on an as needed basis and can easily avoid the capital costs of software and hardware[1].

Cloud computing gets its name as a metaphor for the internet. Economically, the main appeal of cloud computing is that customers only use what they need, and only pay for what they actually use. Resources are available to be accessed from the cloud at any time, and from any location via the internet. There is no need to worry about how things are being maintained behind the scenes. The job of the customers is to just purchase the service they need from the cloud Because of this, cloud computing has also been called utility computing, or IT on demand'. The convenience and efficiency of this approach, however comes with privacy and security risks. A major feature of the cloud services is that users' data are usually processed remotely in unknown machines that users do not own or operate. While enjoying the convenience brought by this new emerging technology, users' fears of losing control of their own data (particularly, financial and health data) can become a significant barrier to the wide adoption of cloud services. In cloud data storage system, users store their data in the cloud and no longer possess the data locally. Thus, the correctness and availability of the data files being stored on the distributed cloud servers must be guaranteed. One of the key issues is to effectively detect any unauthorized data modification and corruption, possibly due to server compromise and/or random failures. Besides, in the distributed case when such inconsistencies are successfully detected, to find which server the data error lies in is also of great significance, since it can always be the first step to fast recover the storage errors and/or identifying potential threats of external attacks [3]. Moving data into the cloud offers great convenience to users since they don't have to care about the complexities of direct hardware management.

The pioneers of cloud computing vendors are Amazon Simple Storage Service (S3) and Amazon elastic Compute Cloud (EC2) are both well known cloud service providers[2]. While these internet-based online services do provide huge amounts of storage space and customizable computing resources, this computing platform shift, however, is eliminating the responsibility of local machines for data maintenance at the same time [3]. As a result, users are at the mercy of their cloud service providers for the availability and integrity of their data. However the fact that the users no longer have physical possession of data in the cloud prohibits the direct adoption of traditional cryptographic primitives for the purpose of the data integrity protection. Hence the verification of cloud storage correctness must be conducted without explicit knowledge of whole data files. Last but not the least, the deployment of cloud computing is powered by data centres running in a simultaneous, cooperated and distributed manner. It is more advantage for individual users to store their data redundantly across multiple physical servers so as to reduce the data integrity and availability threats. Thus distributed protocols for storage correctness assurance will be of most importance of achieving

ISSN: 2278 – 1323

robust and secure cloud storage systems. Data owner should not bother about his data, and should not get fear about damage of his data by hacker; there is need of security mechanism which will track usage of data in the cloud. It is essential to provide an effective mechanism for users to monitor the usage of their data in the cloud. For example, users need to be able to ensure that their data are handled according to the service level agreements made at the time they sign on for services in the cloud.

Accountability is necessary for monitoring data usage, in this all actions of users like sending of file are cryptographically linked to the server, that performs them and server maintain secured record of all the actions of past and server can use the past records to know the correctness of action. It also provides reliable information about usage of data and it observes all the records, so it helps in make trust, relationship and reputation. So accountability is for verification of authentication and authorization. It is powerful tool to check the authorization policies [5]. Accountability describes authorization requirement for data usage policies.

The methodology will also take concern of the JAR file by converting the JAR into obfuscated code which will adds an additional layer of security to the infrastructure. Apart from that we are going to enlarge the security of user's data by provable data possessions for integrity verification. Based on the configuration settings defined at the time of creation, the JAR will give usage control associated with logging, or will give only logging functionality. As for the logging, every time there is an access to the data, the JAR will automatically produce a log record.

II.LITERATURE SURVEY

In this section review related works addressing security in cloud. Security issue is very important in cloud there are many techniques available so here is review of all these.

S. Pearson et al describes privacy manager mechanism in which user's data is safe on cloud , in this technique the user's data is in encrypted form in cloud and evaluating is done on encrypted data, the privacy manager make readable data from result of evaluation manager to get the correct result. In obfuscation data is not present on Service provider's machine so there is no risk with data, so data is safe on cloud, But this solution is not suitable for all cloud application, when input data is large this method can still require a large amount of memory[6]. In [3], the authors present procedural and technical solution both are producing solution to accountability to solving security risk in cloud in this mechanism these policies are decided by the parties that use, store or share that data irrespective of the jurisdiction in which information is processed. But it has limitation that data processed on SP is in unencrypted at the point of processing so there is a risk of data leakage. In [11], the author gives a language which permits to serve data with policies by agent; agent should prove their action and authorization to use particular data. In this logic data owner attach Policies with data, which contain a description of which actions are allowed with which data, but there is the problem of Continuous auditing of agent, but they provide solution that incorrect behaviour. Should monitor and agent should give justification for their action, after that authority will check the justification. In [7], authors gives a three layer architecture which protect information leakage from cloud, it provides three layer to protect data, in first layer the service provider should not view confidential data in second layer service provider should not do the indexing of data, in third layer user specify use of his data and indexing in policies, so policies always travel with data. In [8], authors present accountability in federated system to achieve trust management. The trust towards use of resources is accomplished through accountability so to resolve problem for trust management in federated system they have given three layers architecture, in first layer is authentication and authorization in this authentication does using public key cryptography. Second layer is accountability which perform monitoring and logging. The third layer is anomaly detection which detects misuse of resources. This mechanism requires third party services to observe network resources.

III.EXISTING SYSTEM

The primitive security techniques like indexing, policy enforcement, java based techniques, and Authentication based techniques are used. In indexing technique there is a violation of access control, which is not tolerable in cloud computing. In policy enforcement mechanism, some policies can be easily violated by the user and some policies can also be compromised by the attacker so this mechanism is not so efficient. In case of java based techniques, the objects called Self defending Objects are used, this technique is better compared to policy enforcement mechanism, but Self Defending Objects does not provide total security to the cloud.

Last but not the least the authentication techniques, in these there are two types of authentication is used they are PCA(Principal Component Analysis) and IBE(Identity Based Encryption)[9], among this two IBE is used till today in cloud, although it is a primitive security technique, it is reliable and efficient. As a result of advancement in the technology, the hacking techniques are also improved. So in order to ensure total security for the cloud and its resources, a new system is designed.

Conventional access control approaches made for closed domains such as databases and operating systems, or approaches with a centralized server in

ISSN: 2278 – 1323

distributed environments are not suitable, because of the following features characterizing cloud environments.

IV.PROPOSED SYSTEM

To overcome the above problems, we propose a novel method, namely Cloud Information Accountability (CIA) framework, based on the notion of information accountability. Data Owner can upload the data into the cloud server after encrypted the data. User can subscribe into the cloud server with certain access polices such as read, write and copy of the original data. The JAR programmable capabilities are included to create both a dynamic and travelling object, and to ensure that any access to users 'data will to the trigger authentication and automated logging local JARs. To strengthen user's control, a distributed auditing mechanism has been incorporated. The Loggers and Log Harmonizer will have a track of the access logs and reports to the data owner. This Process ensures security. There is a provision of extensive experimental studies that demonstrate the efficiency and effectiveness of the proposed approaches with the following constraints.

1. The logging should be decentralized in order to adapt to the dynamic nature of the cloud. More specifically, log files should be tightly bounded with the corresponding data being controlled, and require minimal infrastructural support from any server.

2. Every access to the user's data should be correctly and automatically logged. This requires integrated techniques to authenticate the entity that access the data, verify, and record the actual operations on the data as well as the time that the data have been accessed.

3. Log files should be reliable and tamper proof to avoid illegal insertion, deletion, and modification by malicious parties. Recovery mechanisms are also desirable to restore damaged log files caused by technical problems.

4. Log files should be sent back to their data owners periodically to inform them of the current usage of their data. More importantly, log files should be retrievable anytime by their data owners when needed regardless the location where the files are stored.

5. The proposed technique should not intrusively monitor data recipient's systems, nor it should Introduce heavy communication and computation overhead, which otherwise will hinder its feasibility and adoption in practice.

V. DISTRIBUTED ACCOUNTABILITY AND AUDITING

A. CIA FRAMEWORK

The innovative features of the CIA framework[10] lies in its ability in maintaining light-weight and powerful accountability that combines aspects like access control and usage control. This CIA framework helps the data owners in tracking whether the service level agreements are honoured or not and it also gives a additional provision for the data owners to enforce access and usage control rules. Associated with the accountability feature, two distinct modes for auditing is designed namely push mode and pull mode. The push mode refers to the logs periodically sent to the data owner, and the pull mode refers to an alternative approach where the user can retrieve the logs when needed.

The CIA framework also conducts automated logging and auditing mechanisms by using two major components they are logger and log harmonizer. The logger is strongly coupled with the user's data, its job is to automatically record the logging access to data items that it contains and send the encrypted log records to the log harmonizer. The logger can also be configured to ensure the access and user control policies are honoured. The logger requires minimal server support from the server; as a result it is not a burden for the server to process along with the data. The tight coupling between the data and the logger, results in a highly distributed logging system. The second component of the CIA framework is the log harmonizer, this is responsible for auditing and includes two auditing strategies push mode and pull mode. The logger and log harmonizer both are enclosed within a JAR file.

The programmable capability of the JAR files is extended in order to automatically log the usage of the user's data by any entity in the cloud. The entire user's data along with their policies such as access control policies and logging policies that they want to enforce, are enclosed within the JAR files and sent to the cloud service providers. Any access to the data will trigger an automated authenticated and authenticated logging mechanism local to the JAR's. This type of enforcement is also termed as strong binding since the policies and logging mechanisms travel with the data [4]. This strong binding exists even when the copies of the JAR's are created, thus the user will have control over his data at any location. This type of decentralized logging mechanism meets the dynamic nature of the cloud. The JAR files are connected with a central point of contact, which forms a link between them and the user. It records the error correction information sent by the JARs which facilitates it to monitor the loss of any logs from any of the JARs. If there is a problem for a JAR to make contact with the central point, then

ISSN: 2278 – 1323

access to its enclosed data is denied. The architecture of the CIA framework is shown.

The figure 1 depicts a complete architecture of the CIA framework. Data owner, cloud testbed and users are some of the important entities in the architecture. In order to initiate the communication between the data owner and the cloud a network certificate is generated. This is used as a handshaking between the cloud server and the data owner.

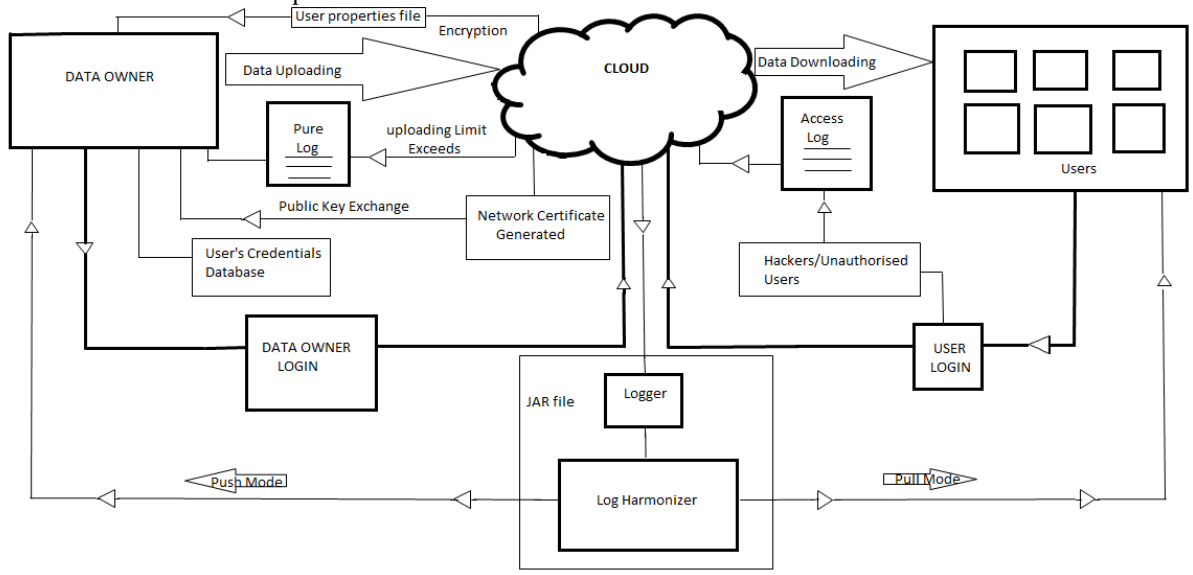

Fig 1: Architecture of the Cloud Information Accountability Framework

B. *MODULES*

DATA OWNER: Data Owner is the Person who is going to upload the data in the Cloud Server. In order to upload the data into the Cloud server, the Data Owner have to be registered in the Cloud Server. Once the Data Owner registered in cloud server, the space will be assigned to the Data Owner.

USER: A user plays a vital role in cloud computing, for whom the whole system is dedicated and designed. A new user should register with the data owner, to access the data owner files stored in the cloud server. Once he is registered, then he becomes an authorized user and he can send a file request to the data owner. As soon as the data owner receives a request, he verifies the user's credentials, if they are valid then the user should send a request for the file key, which is second level of security check, along with the request the user is again needed to send his name and also the file name, which he needs to access or download, again his name is cross verified by the data owner and a file key is sent for the requested file. So, once the user receives the file key he can download the requested file by entering the received file key along with the payment details, all these entries are verified by the data owner and the requested file allowed to be downloaded by the user.

CLOUD (Cloud Service Provider): Cloud Server is the area where the user going to request the data and also the data owner will upload their data. Once the user send the request regarding the data they want, the request will be first send to the Cloud Server and the Cloud Server will forward your request to the data owner. The data Owner will send the data to the user via Cloud Server. The Cloud Server will also manage the Data owner and Users information in their Database for future purpose.

LOGGER: The Logger is maintained by the Cloud Server. Loggers have the details of the data owner and users who are accessing the Cloud Server. So the Logger will be more useful for many purposes. Like which user / data owner accessing the Cloud Server, accessed at the particular time and the IP address from which the data is requested by user etc.

NETWORK CERTIFICATE GENERATOR: The Certificate Generator is used to verify the Cloud Server is recognized or not. The Cloud Server has to be recognized by the certificate generated. If not recognized, the Cloud Server is a Fraudulent Server. The data owner can check whether the recognized or not because the data owner is going to upload their data in the Cloud Server.

LOG HARMONIZER: This is responsible for auditing, the log harmonizer has the capability of handling the logging mechanism also, but to decrease the workload it is handled by the logger and separating logging and auditing mechanism increases performance. We can infer that, it supports two auditing strategies like pull mode and push mode.

PULL MODE: In the Pull method, the data owner has to send the request to the Cloud Server

ISSN: 2278 – 1323

regarding the access details of their data up to the particular time. Then the Cloud Server will send

PUSH MODE: For the every periodical time the Cloud Server will send the access details of the user to the data owner. So that the Data Owner may able to know who're all the accessing their data at the particular time period. During the registration phase, the Data owner will ask by the Cloud Server whether they're choosing the push or pull method.

PURE LOG: Pure log is the log which is reserved only for the access and benefit of the data owner. The pure log is updated when the data owner uploads the file, the working of pure log can be explained in a simple way that is when a data owner uploads a file to the cloud which exceeds the uploading file limit size, then immediately a error message is displayed to the owner and a log record is generated in the pure log.

ACCESS LOG: Access log is the log which is designed and maintained to ensure complete security for the user's files stored in the cloud server. This access log can be retrieved by both the user and data owner whenever needed. Access log helps in tracking the hacker's location easily and the requests from the particular ip address can also be blocked.

VI. COMPLETE DATA FLOW

Primarily the data owner before uploading the file in the cloud, he should get registered with the cloud service provider, so he should create an account in the cloud, hence a request is sent by the data owner to the cloud service provider. Once the data owner creates a account in the cloud, the cloud server sends account acceptance response, so from this instance of time the data owner can start uploading his files to the cloud, immediate messages will be displayed to the data owner after each and every successful file upload, this is nothing but the file reception confirmation sent from the cloud server.

Similar type of communication is carried between the user and the cloud server, primarily a new user should create a account, after creating a account, the confirmation will be sent from the cloud server to the user. Now the user gets the privilege of downloading the files stored in the cloud. So if a user wants to download a file, then a file request is sent by the user to the cloud server, after verifying the user's credentials, the cloud server sends file request confirmation and also allows the user to download the requested file.

VII.WORKING OF LOGGING AND AUDITING MECHANISM

In CIA framework logging and auditing mechanisms plays a vital role in ensuring total security. The term logging means binding the JAR file strongly to the user's data and also to the files stored in the cloud. This strong binding is necessary so that, any access to the user's data or to the data owner files would immediately trigger the logging mechanism[11] and the access log gets updated automatically. Hence logging is very much essential in proper working of the whole mechanisms in the cloud. Moving on to the auditing technique, auditing is very much necessary, without which logging technique remains incomplete. Auditing can be defined as sending the access log through push and pull mode to data owner and the user respectively.

THE LOG RETRIEVAL ALGORITHM FOR GENERATING PURE LOG

```
1: Let TS (NTP) be the Network time protocol
timestamp
2: pull=0
3: rec: = (Owner name, File name, File size,
Timestamp)
4: curtime: = TS (NTP)
5: lsize: = sizeof(log) //current size of the log
6: if ((curtime –tbeg) < time)&&(lsize <
size)&&(pull==0) then
7: log: =log + (rec)
8: PING to harmonizer // send a ping to harmonizer
to check if it is alive
9: if PING-harmonizer then
10: PUSH RS(rec) // write the error correcting bits
11: else
12: EXIT(1) // error if no ping is received
13: end if
14: end if
15: if ((cutime – tbeg) > time)||(lsize >= size)||(pull
0) then
16: // Check if PING is received
17: if PING-harmonizer then
18: PUSH log // write the log file to the harmonizer
19: RS(log) := NULL // rest the error correction
records
20: tbeg := TS(NTP) // reset the tbeg variable
21: pull := 0
22: else
23: EXIT(1) // error if no PING received
24: end if
25:end if
```

In the above algorithm size keyword is used to indicate the size of the log file specified by the data owner, time keyword indicates the maximum time allowed to elapse before the previous log generation, tbeg keyword indicates the timestamp at which the last log was recorded. The above algorithm presents logging and synchronization steps with the harmonizer in case of pure log.

ISSN: 2278 – 1323

Primarily the algorithm checks whether the size of the file has exceeded a stipulated size, if the condition fails then the pure log is not generated or else a pure log record is generated.

AN ALGORITHM FOR GENERATING ACCESS LOG
1: Let TS (NTP) be the network time protocol timestamp.
2: pull=0
3: rec:=(Hackers name, File Name, Owner Name, Time stamp, IP address)
4: curtime:=TS (NTP)
5: lsize:=size of (log) // current size of the log
6: if ((curtime-tbeg)<time)&&(lsize<size)&&(pull==0)then
7: //Check if PING is received.
8: log: = log+ (rec)
9: PING to harmonizer // send a PING to the harmonizer
10: if PING-harmonizer then
11: PUSH RS (rec) // write error correcting bits
12: else
13: EXIT (1) // error if no PING is received
14: endif
15: endif
16: if ((curtime-tbeg) > time)||(lsize=size)||(pull 0) then
17: //Check if PING is received
18: if PING-harmonizer then
19: PUSH log // write the log file to the harmonizer
20: RS (log):=NULL // reset the error correction records
21: tbeg:=TS (NTP) // reset the tbeg variable
22: pull:=0
23: else
24: EXIT (1) // error if no PING is received
25: endif
26: endif

The above algorithm is used for retrieving access log, an additional check is added after the step 6. Precisely the access log checks whether the user is an authorized person and satisfies all the conditions specified in the policies pertaining to it. If the conditions are satisfied, access is granted otherwise, access is denied and the access log, logs the user as a hacker.

VIII.SECURITY DISCUSSION

The CIA framework is a strong shield against some of the deadly security attacks like Man- in –Middle attack, Compromising JVM attack, Attacks on JAR files and disassembling attack[4].

A. Attacks on JAR files:
The common attack that we can assume is accessing the data in JAR file without being noticed. But such attack can be found out by auditing. However if someone tries to download the JAR files, the actions are recorded by the logger and the log record is sent to the user. By this the data owner will be aware of his/her JAR file download.

B. Compromising JVM attack:
In case of compromising JVM attacks, where JVM is acronym for JAVA Virtual Machine, the solution relies on the correctness of the JVM. If the JVM is compromised, all the code executed on it can be compromised. These kinds of attacks can be prevented in several ways, one approach is to verify the integrity of a virtual machine prior to the running of JAR‘s, for which the data owner needs to check whether the image of the virtual machine being loaded into the memory matches that of the provider. Another way is before opening the JAR, a command can be included in the script to repair the JVM using the version provided by the sun Microsystems. If the cloud is above an untrusted network, a JRE can be used as a package along with the JAR‘s.

C. Man-in- Middle Attack:
This attack can be stated as an attacker may intercept messages during the authentication of a service provider with the certificate authority, and reply to the messages in order to masquerade as a legitimate service provider. There are two situations where the attacker can reply for the messages, one is after the actual service provider has completely disconnected and ended a session with the certificate authority. The other is when actual service provider is disconnected but the session is not over, so the attacker may try to renegotiate the connection. The first type of attack will not succeed since the certificate typically has a timestamp which will become obsolete at the time point of reuse. The second type of attack will also fail since renegotiation is banned in the latest versions of network certificates and cryptographic mechanisms.

D. Disassembling Attack
In this attack, the attacker disassembles the JAR file of the logger and then attempt to extract useful information out of it or spoil the logs in it. In first case the attacker may try to identify which encrypted log records corresponds to this actions by making a plaintext attack to obtain some pairs of encrypted log records and plain texts. But the CIA framework is enclosed by a Weil pairing algorithm which ensures that the framework has both cipher text as well as plaintext security. So it is impossible for the attacker to decrypt the data or the log files in the disassembled JAR file.

ISSN: 2278 – 1323

IX. CONCLUSION

This is an attempt to enhance and improve the security of the cloud both internally and externally. The main goal of this work is to protect the information stored in the cloud server and also to protect the cloud resources from unauthorized access. This approach involves total security for both the user‘s and owner‘s data and also involves automatic logging and auditing mechanisms. There may be many works has been done in the field of cloud security, but in this work different techniques and different mechanisms are used, and also by keeping in mind that the time and resources consumed by these security aspects are minimum and can be easily handled by the server. This approach allows the data owner to not only audit his content but also enforce strong back end protection and control over the file access policies.

In this approach major importance is given to the security aspects of the cloud. As an enhancement from security perspective in future we can design a system that sends alerts to the cell phones, when there is a security violation, and an additional feature also has been included in this work, that is blocking the IP of the attacker. This is one of the simple and effective mechanisms because the IP of the attacker/hacker will be recorded in the access log, so the data owner can block the particular IP just with a button click. Once the IP is blocked then the hacker cannot get into the cloud.

REFERENCES

[1] Cloud Computing, Principles and Paradigms by John Wiley & Sons.

[2] 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.

[3] S. Pearson and A. Charlesworth, “Accountability as a Way Forward for Privacy Protection in the Cloud,” Proc. First Int’l Conf. Cloud Computing, 2009.

[4] Ensuring Distributed Accountability for Data Sharing in the Cloud Author, Smitha Sundareswaran, Anna C.Squicciarini, Member, IEEE, and Dan Lin, IEEE Transactions on Dependable and Secure Computing ,VOL 9,NO,4 July/August 2012 .

[5] D.J. Weitzner, H. Abelson, T. Berners-Lee, J. Feigen-baum, J. Handler, and G.J. Sussman, “Information Accountability”, Comm.ACM,vol. 51,no. 6,pp. 82-87,2008

[6] S. Pearson , Y. Shen, and M. Mowbray," A privacy Manager for Cloud Computing," Proc. Int'l Conf. Cloud Computing (cloudcom), pp.90-106,2009.

[7] A. Squicciarini , S. Sundareswaran and D. Lin, " Preventing Information Leakage from Indexing in the Cloud," *Proc. IEEE Int'l* Conf. Cloud Computing, 2010.

[8] B. Chun and A. C. Bavier ,"Decentralized Trust Management and Accountability in Federated System," *Proc. Ann. Hawaii Int'l Conf. System Science (HICSS), 2004.*

[9] D. Boneh and M.K. Franklin, “Identity-Based Encryption from the Weil Pairing,” Proc. Int’l Cryptology Conf. Advances in Cryptology, pp. 213-229, 2001.

[10] Nilutpal Bose, Mrs. G. Manimala, “SECURE FRAMEWORK FOR DATA SHARING IN CLOUD COMPUTING ENVIRONMENT, Website: www.ijetae.com, Volume 3, Special Issue 1, January 2013)

[11] R. Corin, S. Etalle, J.I. den Hartog, G. Lenzini, and I. Staicu, “A Logic for Auditing Accountability in Decentralized Systems,” Proc. IFIP TC1 WG1.7 Workshop Formal Aspects in Security and Trust, pp. 187-201, 2005.

CHANDAN.M is an M.Tech student of Rao Bahadur Y Mahabaleswarappa Engineering College, Bellary, India. Presently he is pursuing his M.tech(CSE) from this college and he received his B.tech from Channabasaveshwara InStitute Of Technology, Tumkur. His area of interest includes Cloud Computing and Object Oriented Programming languages, all current trends and techniques in Computer Science

APARNA K S, Asst-Professor, Dept of CSE, Rao Bahadur Y Mahabaleswarappa Engineering College, Bellary, India.

ISSN: 2278 – 1323