ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, *Volume 3, Special Issue 1, February 2014*

International Conference on Engineering Technology and Science-(ICETS'14)

On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

A Secure Data Distribution and JAR Authentication in Cloud Storage Systems

Mr.PREMKUMAR.M.G[1], Mr. K.SUDHAKAR[2], Ms.R.MENAKA[3]

IV SEM,M.E CSE ,Department of CSE, Sengunthar College Of Engineering, Trichengode, India[1]
Assistant prof , Department of CSE, Sengunthar College Of Engineering, Trichengode, India[2]
Assistant prof. [Sl.Gr], Department of IT, Velalar College Of Engineering And Tech, Erode, India[3]

Abstract: Cloud computing enables highly scalable services to be easily consumed over the Internet on an as-needed basis. 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. To address this problem, we propose a Decentralized information accountability framework. We propose a logging mechanism along with user's data and policies. Whenever a user logs into the system, then a log record will be generated and those log records will be stored in a JAR (Java Archives) file. We propose a single JAR which has two functionalities like matching policies and providing/decrypting the file if that policy is matched. One more approach we are providing ABE technique which creates attributes and ensures security for user's related information. In addition to this we proposed a JAR signer tool used for signing and verification of JAR file. We provide extensive experimental studies that demonstrate the efficiency and effectiveness and the proposed approaches.

Keywords-Cloud computing; Accountability; ABE; JAR signer

I. INTRODUCTION

Cloud computing offers highly scalable services are consumed trough internet. A significant barrier to the adaptation of cloud services is thus the user fear of losing his own data and loss of privacy in the cloud. There are number of cloud computing services including Amazon, Google, Microsoft, Yahoo and Sales-force. The services which are abstracted by the user are no need to be the experts of technology infrastructure. This new technology is beneficial but users will start worrying about losing control of their own data. Governance and accountability questions are the main privacy issue of cloud computing.

Accountability plays a critical role in the development of trust during human interaction. Thus, accountability is viewed both as a tool to achieve practical security and as a first class design goal of services in federated distributed systems. There are two types of accountability Agent accountability and Data accountability. Agent accountability focuses on whether the actions of a given agent where authorized. For single data usage policy the data accountability describes the authorization requirements. In our paper we mainly concentrate on data accountability. Accountability will be achieved via a combination of private and public accountability. Public accountability is a active interaction between subjects of personal identifiable information, regulatory bodies, such as data controllers. Private accountability, in contrast is derived from the interaction between data controllers and data processors. For two reasons the approaches using a centralized server is not suitable. First, outsourcing of data by cloud service provider to storing data. Outsourcing of data into the cloud reduces the cost and complexity of long term large scale data storage, but it does not offer guarantee on data integrity and availability [2]. So that here proposed a trusted computing environment that provides secure cross platform. Authentication, encryption and decryption, compression are the security issues. Second, there will be a flexible manner of entities.

To overcome these problems, a new approach has been proposed namely Cloud Information Accountability(CIA). Here it uses Java Archives (JAR) files for automatically log the usage of user's data, it can provide distributed auditing mechanism and also can provide authentication of JAR for the

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)

On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

strengthening of user's control, it develops powerful application even after they modify the code and the code of the copied code by the attacker. CIA framework combines the feature of access control, usage control and authentication. In CIA framework they have used a Java Archives (JAR) files which will automatically log t usage of the user data. JAR file includes user's data and their policies. By using this framework it helps to increase the trustiness of cloud. Cloud storage must be handling with full care of security. Cloud security can include access control, usage control and authentication. Considering the above related works cloud security can be provided by using many methods. Among that the most efficient mechanism is providing accountability for cloud data. By using CIA framework it trace the control of data by the data owner. Here the user, who subscribed to a certain cloud service, usually needs to send his data as well associated access control policies to the service provider after the data are received by the cloud service provider, the granted access is given to service provider, such as read, write and copy on the data. Logging and auditing technique is developed to track the data. There are two modes for auditing: push mode and pull mode in the accountability feature. The push mode refers to logs that are periodically sent to data owner. In contrast pull mode refers to retrieving the logs on as a needed basis. We provide the JARs with a central point of contact which forms a link between them and the user in a decentralized logging mechanism. This mechanism also records the error correction information which has been
sent any of the JARs. Suppose, if any of the JAR is not able to contact its central point, then access to its enclosed data will be denied.

In our paper, the contributions are as follows

- ✓ This is the first time we are proposing a usage of JAR files for the accountability and includes a logging mechanism
- ✓ In our architecture we are using decentralized server because it does not require any authentication or storage system.
- ✓ Our architecture is platform independent
- ✓ In addition to access control, we also provide usage control.

This paper is an extension of our previous conference paper. We have made the following new contributions. In order to strengthen the dependability of our system in case of comprised JRE we integrate oblivious MD5 hashing and integrity check. We also updated the log records structure to provide additional guarantees of integrity and authenticity. Second, to recover more possible attack scenario we extended the security analysis. Third, the results of new experiments are reported and provided a thorough evaluation of the system performance.

The rest of the paper is organized by sections: related work will be discussed in section 2. Proposed system is discussed in section 3.Our proposed Implementation is discussed in section 4, Experimental results in section 5, Section 6 describes conclusion and future work.

II. BACKGROUND AND RELATED WORK

Initially we will review related work addressing issues like data sharing in the cloud. User's data are usually
processed in remote machines this is the main feature of cloud service. These unknown machines is not owned by the users and they cannot operate also the users will be afraid of losing control of their data in particular health and financial related data[1]. For the wide adoption of cloud services this can become a significant barrier. To overcome this problem they have proposed accountability framework that is going to keep track of data. They have also proposed JAR programmable framework that will trigger authentication whenever there is an access to the data. There are two main problems that will arise in cloud environment. First is the data which has to be handled will be outsourced [4] and second is entities will enter and exit the cloud in a flexible manner. As a result Handling data becomes complex. To address the above problems, they have proposed Cloud Information Accountability framework. This will keep usage of data transparent. There are two main modes for auditing that is push mode and pull mode. The push mode sends the log to the data owner. Whereas pull mode retrieves that log. The data which has to be send by user also contain policies. Pearson developed privacy manager [2] and he has also proposed the mechanisms for auditing. The layered architecture has been presented for end to end trust management addressing .In the electronic commerce they investigated accountability as a provable property by the researchers. Usage of policies is proposed by the authors .Accountability based distributed system is designed by the Jagadeesan The only work proposing a distributed approach to accountability is from Lee and

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)

On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

colleagues. Self-defending objects are related to our work. The java based approach has been proposed to prevent privacy leakage from indexing. Proposed an access control and for strongly coupling content using Attribute Based Encryption (ABE). For secure data provenance our work looks similar but greatly differs in terms of goals, techniques and application domains. Investigation of usage control is carried out for the extension of current access control mechanisms. They mainly focused on conceptual analysis. Constraints at various level of granularity. These are the issues of privacy and security.

III. PROPOSED WORK

The main aim of the proposed system is to design an accountability framework for modeling a highly scalable cloud environment. This will enables policies in a JAR file. In this project work, the focus is on presenting the proposed technique, a scalable, reliable service model for policies in cloud that uses a logging mechanism and also providing security for the same. It has to show that the proposed system saves the time overhead, in which it won't lose the control of data in the cloud.

The aim to develop logging and auditing technique which satisfy the following requirements.

1. Implement CIA framework allowing user to generate keys.
2. Implement CIA framework allowing user to mention the access control rules.
3. Implement CIA framework to package the data, access rules and a key to a JAR component.
4. Every time user access data, log records must be generated, encrypted and stored in log area.
5. Implement mechanism to keep the log record safe, to avoid missing records and data corruption.
6. Implement a portal for the user to view the access statistics from the log files.
7. Implement JAR Signer tool to verify integrity of JAR file and automatic logging facility of JAR.
8. Implement Md5 hash function for efficient encryption of data.

The method supposes the log record to avoid missing records, thereby it provides security for the successful usage of policy in the cloud. The proposed system illustrates that a data owner who is the owner of all related data, accepts a key from a ABE key generator and a logger is responsible for the view of data owner. Logger contains the information of the person who has requested to access the data. The log entities should be initially sent to the logger component. The key which is passed to the data owner will be encrypted using an ABE encryption technique.

The Encrypted key will be sent to the JAR which has been generated. The JAR file is loaded to the JAR executer. The authentication system is provided in which JAR executer will enable authentication request and authentication response to that system. The user of the data will give access to the data if request is valid. The data which is sent to the authentication system will be finally sent to cloud storage provider (Azure). The proposed system illustrates that a data owner who is the owner of all related data, accepts a key from a ABE key generator and a logger is responsible for the view of data owner. Logger contains the information of the person who has requested to access the data. The log entities should be initially sent to the logger component. The key which is passed to the data owner will be encrypted using an ABE encryption technique. The encrypted key will be sent to the JAR which has been generated.

IV. MODULE DESCRIPTION

i) Log Record Generation

Log records are generated by the logger component. Logging occurs at any access to the data in the JAR, and new log entries are appended sequentially, in order of creation LR=<r1… rk> Each record r1 is encrypted individually and appended to the log file. In particular, a log record takes the following form:

r1=<ID Act ,T ,Loc ,h((ID ,Act ,T ,Loc)|ri-1|…|r1), sig):

Here, r1 indicates that an entity identified by ID has performed an action Act on the user's data at time T at location Loc. The component h((ID, Act ,T ,Loc{ri-1|…|r1) corresponds to the checksum of the records preceding the newly inserted one, Where ID indicates identity. Concatenated with the main content of the record itself . The checksum is computed using a collision-free hash function. The component sig denotes the signature of the record created by the server. If more than one file is handled by the same logger, an additional Obj ID field is added to each record.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

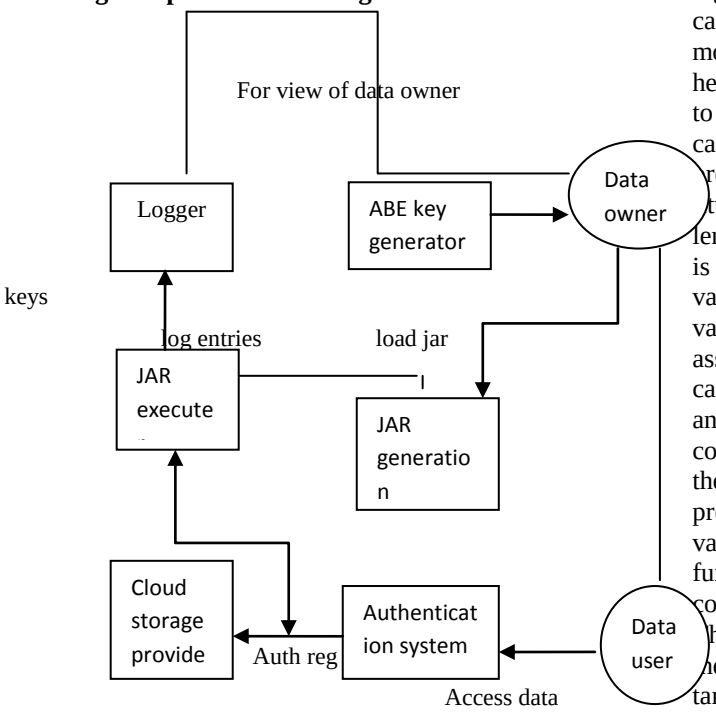
An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)

On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

Fig1: implementation diagram



ii) JAR File Generation

The JAR file is generated in which both creates a dynamic and travelling object and to ensure that any access to the user data will trigger authentication and automated logging local to the JAR. These JARs will automatically log the usage of the user data by any entity in the cloud. The access policies will be sent by the users enclosed in JAR file to cloud service provider. The user will create a logger component which is a JAR file, to store its data items using a generated key. The JAR file includes a set of simple access control rules specifying whether and how the cloud server and possibly other data stakeholders (users, companies) are authorized to access the content itself. Then, the JAR file will be sent to the cloud service provider that he subscribes to authenticate CSP to JAR, we use open SSL based certificate.

iii) Hash Function Creation

A hash function is any algorithm or subroutine that maps large data sets of variable length to smaller data sets of a fixed length. The values returned by a hash function are called hash values, hash codes, hash sums, checksums or simply hashes. Hash value plays an important role in correcting the logs. Initially, the hash codes are inserted, which calculate the hash values of the program traces of the modules being executed by the logger component. This helps us detect modifications of the JRE, and are useful to verify if the original code. The integrity checks are carried out using MD5 Hashing. The input message is broken up into chunks of 512-bit blocks (sixteen 32-bit little endian integers),the message is padded so that its length is divisible by 512 executed. The hash function is initialized at the beginning of the program, the hash value of the result variable is cleared and the hash value is updated every time there is a variable assignment, branching, or looping. The hash code captures the computation results of each instruction and computes the oblivious-hash value as the computation proceeds. These hash codes are added to the logger components when they are created. They are present in their JAR. The log harmonizer stores the values for the hash computations. The main functionality of log harmonizer is it forms the central component which allows the user access to log files. The values computed during execution are sent to it by the logger components. To verify the JRE has been tampered, the log harmonizer proceeds to match these values against each other.MD5 is a message digest algorithm developed by Ron Rivest at MIT. It is basically a secure version of his previous algorithm, MD4 which is a little faster than MD5. This has been the most widely used secure hash algorithm particularly in Internet-standard message authentication. The algorithm takes as input a message of arbitrary length and produces as output a 128-bit message digest of the input. This is mainly intended for digital signature applications where a large file must be compressed in a secure manner before being encrypted with a private (secret) key under a public key cryptosystem.

(1)Padding :The message is padded to ensure that its length in bits plus 64 is divisible by 512. That is, its length is congruent to 448 modulo 512. Padding is always performed even if the length of the message is already congruent to 448 modulo 512. Padding consists of a single 1-bit followed by the necessary number of 0-bits.

(2) Appending length: A 64-bit binary representation of the original length of the message is concatenated to the result of step (1). (Least significant byte first). The expanded message at this level will exactly be a multiple of 512-bits. Let the expanded message be

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

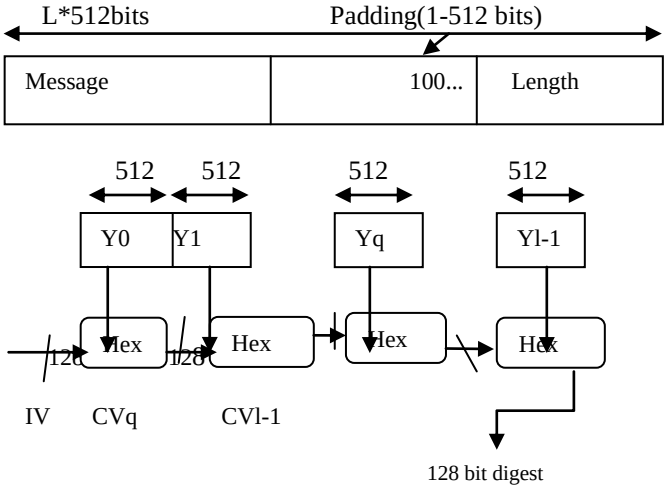
International Conference on Engineering Technology and Science-(ICETS'14)

On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

represented as a sequence of *L* 512-bit blocks *Y0, Y1,..,Yq,..,YL-1*.
Note that in the figure, IV and CV represent initial value and chaining variable respectively.

Fig 2:Message digest



(3) Initialize the MD buffer: The variables IV and CV are represented by a four–word buffer (ABCD) used to compute the message digest. Here each A, B, C, D is a 32-bit register and they are initialized as IV to the following values in hexadecimal. Low-order bytes are put first.

Word A: 01 23 45 67
Word B: 89 AB CD EF
Word C: FE DC BA 98
Word D: 76 54 32 10

(4) Process message in 16-word blocks :This is the heart of the algorithm, which includes four "rounds" of processing. It is represented by HMD5 in Figure 1 and its logic is given in Figure 2. The four rounds have similar structure but each uses different auxiliary functions *F, G, H* and *I*.

F(X,Y,Z)=(X ^ Y)∨(X ^ Y)
G(X,Y,Z)=(X ^ $\overline{Z}$) ∨(Y ^ Z)
H(X,Y,Z)= X □ Y □ Z
I(X,Y,$\overline{Z}$)=Y □ □ □ □ X Z)

where □ , □ , □□ and ¯ represent the logical OR, AND, XOR and NOT operations, respectively. Each round consists of 16 steps and each step uses a 64-element table *T* [1 ... 64] constructed from the sine function. Let *T*[*i*] denote the *i*-th element of the table, which is equalto the integer part of 232 times abs(sin(*i*)), where *i* is in radians. Each round also takes as input the current 512- bit block (*Yq*) and the 128-bit chaining variable (CVq).

An array *X* of 32-bit words holds the current 512-bit *Yq*. For the first round the words are used in their original order. The following permutations of the words are defined for rounds 2 through 4:

2(*i*) = (1+ 5*i*) mod 16
3(*i*) = (5+ 3*i*) mod 16
4(*i*) = 7*i* mod 16

(5) Output: After all *L* 512-bit blocks have been processed, the output from *L*th stage is the 128-bit message digest. operations involved in a single step. The additions are modulo 232. Four different circular shift amounts (s) are used each round and are different from round to round. Each step is of the following form

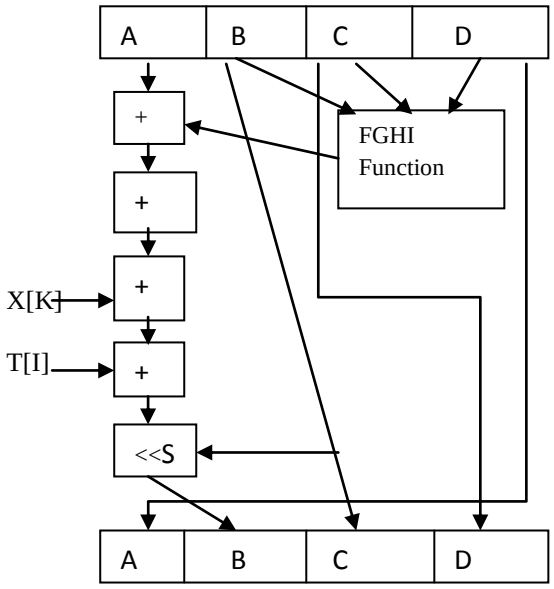


FIG3: operations of MD5

iv) ABE Algorithm

It is a type of public key Encryption in which the public key of a user and the cipher-text are dependent about attributes (e.g. the country he lives, the kind of subscription he has,). In a system the decryption of a cipher-text is possible only if the set of attributes of the user key matches the attributes of the cipher-text. Attribute based encryption can be used for log encryption. Instead of encrypting each part of a log with all the key of all the recipients, it is possible to encrypt the log only with attributes which match recipient's attributes. This primitive can also be used for broadcast encryption in order to decrease the number of key used. Each private key is associated with an access structure that specifies which type of cipher-texts the key can decrypt. We call such a

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)

On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

scheme a Key-Policy Attribute-Based Encryption (KP-ABE). An (Key-Policy) Attribute Based Encryption scheme consists of four algorithms.

Setup Attributes: This algorithm is used to set attributes for users. This is a randomized algorithm that takes
no input other than the implicit security parameter. It defines a bilinear groupG1 of prime order *p* with a generator *g*, a bilinear map *e* : G1×G1→G2 which has the properties of bi-linearity, computability, and nondegeneracy. From these attributes public key and master key for each user can be determined. The attributes,
public key and master key are denoted as

Attributes-U ={ 1,2. . . N}
Public key-PK = (Y, T 1, T 2, . . . , T N)
Master key-MK = (y, t 1, t2, . . . , t N)

Where T I ∈ G1 and ti∈ Z*p* are for attribute i, 1≤ *i* ≤ N , and Y ∈G2 is another public key component. We have
T i=gti and Y =e(g, g)y, y∈ Z*p*. While PK is publicly known to all the parties in the system, MK is kept as a secret by the authority party.

Encryption: This is a randomized algorithm that takes a message M , the public key PK , and a set of
attributes I as input. It outputs the cipher text E with the following format:

E = (I, E, {E i ∈ I)

Where E=MY s,E i=T *is.* And s is randomly chosen from Z.

Secret key generation: This is a randomized algorithm that takes as input an access tree T, the master key MK, and the public key K. It outputs a user secret key SK as follows. First, it defines a random polynomial pi(x) for each node I of T in the top-down manner starting from the root node r. For each non-root node j, pj(0) =p parent (j)(idx(j)) where parent (j) represents j's parent and idx (j) is j's unique index given by its parent. For
the root node r ,pr (0) =y. Then it outputs SK as follows.

SK ={sk i }i ∈ L

Where L denotes the set of attributes attached to the leaf nodes of T and sk i=g pi(0)/ti.

Decryption: This algorithm takes as input the cipher text **E** encrypted under the attribute set **I**,the user's secret key **SK** for access tree **T** , and the public key **PK** . It first computes e(E i,sk i)=e(g, g)pi(0)s for leaf nodes. Then, it aggregates these pairing results in the bottom-up manner using the polynomial interpolation technique. Finally, it may recover the blind factor Y s =e(g,g) ys and output the message **M** if and only if **I** satisfies **T**

Role of ABE in our Proposed System

In our proposed system, we are making use of Key policy based ABE. In earlier work, they have implemented IBE technique which was used to set the time. The main disadvantage of IBE in our system is the attributes has not been defined and the images has sent to all registered users. There was no particular group to set access policies. By making use of ABE technique we are going to define attribute and the users who belong to that particular attribute can access the images, others who are not under that particular category cannot access
images. By using ABE technique in our system we are providing security and performance will get increased when compare to earlier work.

v) JAR signer tool and implementation:

Name: jar signer - JAR signing and verification tool

Synopsis

jar signer [options] jar-file alias

jar signer -verify [options] jar-file alias

Description: The jar signer tool is used for two purposes:

1. to sign JAR files, and

2. to verify the signatures and integrity of signed JAR files. The JAR feature enables the packaging of class files, images, sounds, and other digital data in single file for faster and easier distribution. A tool named jar enables developers to produce JAR files. A digital signature is a string of bits that is computed from some data (the data being "signed") and the private key of an entity (a person, company, etc.).

2) Implementation: The proposed system is implemented in windows XP or windows 7 with Pentium core i4 processor. The design environment is selected in cloud analyst. By making use of single JAR management will become easier and time consumption will become less drastically. So, in our proposed system instead of outer n inner JARs we are making

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)

On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

use of single JAR. We are defining attributes in our proposed system. The data users will get registered to that particular attribute in which they belong to. Suppose a data owner wants to send an image. That image can be viewed by particular user who belongs to that designated group. Other group of people cannot access that image. Thereby we are achieving confidentiality, reliability, and security. Moreover performance will get increased. A download chart is designed in order to make out which users have accessed data owner files, how many times accessed, download count, date of downloading. All information will be in that chart. It keeps the record of data owner.

V. IMPLEMENTATIONS

The proposed system is implemented in windows XP with Pentium IV processor. The design environment is selected in cloud analyst. By making use of single JAR management will become easier and time consumption will become less drastically. So, in our proposed system instead of outer n inner JARs we are making use of single JAR. We are defining attributes in our proposed system. The data users will get registered to that particular attribute in which they belong to. Suppose a data owner wants to send an image. That image can be viewed by particular user who belongs to that designated group. Other group of people cannot access that image. Thereby we are achieving confidentiality, reliability, security. Moreover performance will get increased. A download chart is designed in order to make out which users have accessed data owner files, how many times accessed, download count, date of downloading. All information will be in that chart. It keeps the record of data owner.

VI.EXPERIMENTAL RESULTS

In the experiments, the log file will be created and the time taken will be examined and also its overhead is measured in the system. The overhead will occur at three points with respect to the time First is during the authentication, second is during encryption of log file and finally during merging those log files. Our architecture is light weight with respect to the storage. The experiment is conducted on cloud analyst tool. It mainly considers three timing characteristics namely response time, processing time and transmission time. First we need to define user bases, number of data centres and at last we need to access one file. Once the file is uploaded, it's going to create a JAR file. Our experimental results show the time taken to create a log file, time taken to create a JAR file and about encrypting that JAR file.

	Avg (ms)	min(ms)	max(ms)
Over all response time	**300.00**	**225.11**	**366.06**
Data processing time	**0.34**	**0.02**	**0.66**

VII. CONCLUSION AND FUTURE WORK

We have proposed an ABE technique and generation of single JAR. By proposing ABE technique the data is secured, cannot be outsourced. We have also achieved scalability, reliability and confidentiality to the users data. By the generation of single JAR time is consumed and management of cloud environment will be easier. Their won't be any leakage of data. We also embedded JAR signer tool to improve the integrity of JAR file.

Future we are planning to improve integrity of JAR file.

REFERENCES

[1].Ammann,P.&Jajodia,S.(1993),"Distributed time stamp generation in planar lattice networks", ACMTrans.onComputerSystems11(3),205225.

[2]. Anna C. Squicciarini, Dan lee, Smith Sundareswaran(2012). "Ensuring Distributed Accountability for Data Sharing in the Cloud" Member, IEEE ,VOL. 9, NO. 4.

[3]. D.J. Weitzner, H. Abelson, T. Berners-Lee, J. Feigen-baum, J.Hendler, an G.J. Sussman(2008)., "Information Accountability," Comm.ACM, vol. 51, no. 6pp. 82-87.

[4].J.Handler,J.Feign Kuingh(2013)"Enabling Data Dynamic and Indirect Mutual Trust for Cloud Computing Storage Systems" Ieee Transactions On Parallel And Distributed Systems Vol:Pp No:99 .
[5].Johnpark, ravi chandu(February 2013)," Enhanced Data Security in Cloud Computing with Third Party Auditor" Volume 3,Issue 2.

[6]. M. Xu, X. Jiang, R. Sandhu, and X. Zhang(2007), "Towards a VMMBased Usage Control Framework for OS Kernel Integrity Protection," SACMAT '07: Proc. 12th ACM Symp. Access Control Models and Technologies, pp. 71-80.

[7]. OASIS(2012.) Security Services Technical Committee, "Security, Assertion Markup Language(saml) 2.0,"

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization, Volume 3, Special Issue 1, February 2014

International Conference on Engineering Technology and Science-(ICETS'14)

On 10th & 11th February Organized by

Department of CIVIL, CSE, ECE, EEE, MECHNICAL Engg. and S&H of Muthayammal College of Engineering, Rasipuram, Tamilnadu, India

http://www.oasis-open.org/committees/tc home.php?wg abbrev=security.

[8]. S. Pearson and A. Charlesworth(2009), "Accountability as a Way Forward for Privacy Protection in the Cloud," Proc. First Int'l Conf. Cloud Computing.

[9]. T. Mather, S. Kumaraswamy, and S. Latif(2009)," Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance (Theory in Practice)", first ed. O' Reilly. pp no..123.

[10]. S. Sundareswaran, A. Squicciarini, D. Lin, and S. Huang(2011.), "Promoting Distributed Accountability in the Cloud," Proc. IEEEInt'l Conf. Cloud Computing,