ACEEE Int. J. on Network Security , Vol. 5, No. 1, January 2014

# Cloud Security and Data Integrity with Client Accountability Framework

Prema Mani[1], Theresa Jose[2], Janahanlal Stephen[3]
[1] Computer Science and Engineering Department, Ilahia College of Engineering and Technology, Kerala, India
[1]Email: premamani1987@gmail.com
[2, 3] Computer Science and Engineering Department, Ilahia College of Engineering and Technology, Kerala, India
[2]Email: theresajos@gmail.com
[3]Email: drlalps@gmail.com

*Abstract*—**The Cloud based services provide much efficient and seamless ways for data sharing across the cloud. The fact that the data owners no longer possess data makes it very difficult to assure data confidentiality and to enable secure data sharing in the cloud. Despite of all its advantages this will remain a major limitation that acts as a barrier to the wider deployment of cloud based services. One of the possible ways for ensuring trust in this aspect is the introduction of accountability feature in the cloud computing scenario. The Cloud framework requires promotion of distributed accountability for such dynamic environment[1]. In some works, there's an accountable framework suggested to ensure distributed accountability for data sharing by the generation of only a log of data access, but without any embedded feedback mechanism for owner permission towards data protection[2].The proposed system is an enhanced client accountability framework which provides an additional client side verification for each access towards enhanced security of data. The integrity of content of data which resides in the cloud service provider is also maintained by secured outsourcing. Besides, the authentication of JAR(Java Archive) files are done to ensure file protection and to maintain a safer environment for data sharing. The analysis of various functionalities of the framework depicts both the accountability and security feature in an efficient manner.**

***Index Terms*—Cloud, JAR, HMAC, MD5, PBE, DSA, Framework.**

## I. Introduction

Cloud computing can be considered to be a model for providing convenient and on-demand network access to a shared pool of resources. It increases capacity or adds capabilities for an organisation without investing in new infrastructure or trained personals, all with higher flexibility [3]. The Cloud architecture generally involves a set of loosely coupled components communicating with each other. The components specifically include a set of front end clients, the data centers or a group of distributed servers which forms the back end. The services of cloud can be defined by cloud computing service models and there are four cloud computing categories based on the services and application provided[3].

With all its flexibility and reliability features cloud computing can introduce certain risks. The lack of trust and confidentiality can pose a major problem. Data security is another critical issue in the case of cloud based services. Challenges also exist there for data integrity.

The users no longer possess the control over the data which makes it extremely challenging in order to protect data confidentiality and to enable a secured data sharing. JAR (Java Archive Files) is a compressed file format that can be used for sharing of data in cloud.

Some authors discuss about the trust management and accountability in federated systems (a special type of distributed database management system)[4].The concepts of this work can be a possible solution to the various issues in the cloud computing scenario. The accountability framework provides an enhanced solution to cloud computing usage. Accountability is a feature providing the transparency of activities or it is a state of being accountable (answerable). The actions in the case of the cloud computing environment must be accountable due to the complexity of service provision [5]. Besides, it must also provide required security for cloud environments. This framework can provide both the accountability and security aspects in a confined and provisioning fashion. The main aim of this framework is to employ simple and sophisticated methods to achieve a highly secured and accountable environment for data sharing in cloud environments.

In the case of cloud, Client side authentication is an unavoidable requirement. This implies the generation of a unique key related to the content of source file in the cloud which has to be associated with the client somehow to enable the client to access the resource, by using MD5(Message Digest 5) algorithm whose structure is shown in Ref.[6] .

Similarly, the JAR files are also viable to security attacks. For the protection of its contents from unauthorized access there should be authentication based on digital signatures[7]. Digital signatures can verify an author and also authenticate the message contents by the DSA(Digital Signature Algorithm) as given in Fig.1.

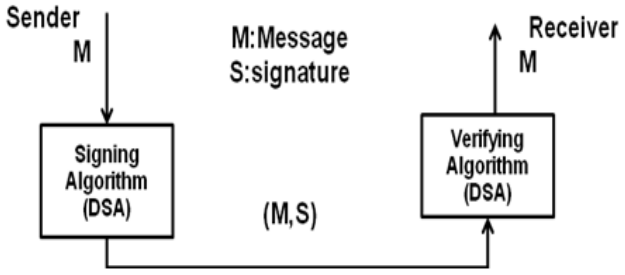


Figure 1. Digital Signature Process


DOI: 01.IJNS.5.1.5

The integrity of data of a particular user must be retained even in the dynamic scenario of cloud .There must be some mechanism to ensure the integrity of data. The mechanism should be to consider the data in a form suitable for analyzing, by the concept of HMAC (Hashed Message Authentication Code) algorithm[8].

It is also necessary to protect the log files in order to verify the accountability and the client access by the data owner through an accountability framework which incorporates password based encryption.

## II. Related Works

The various features related to cloud computing such as cloud service provisioning, scalability and virtualization have been studied and addressed [3].In the case of cloud computing the security and privacy challenges are numerous. The importance of trust management and accountability in federated systems has been dealt with in detail [4]. Data security and privacy issues that can occur once data is placed in the cloud which is accessed from anywhere at any time by the service consumers involving even data modification has been studied [5].

The identification of problem source in the event of data leaks and faults is one of the major sources of uncertainty mainly due to the difficulty in locating the data source, resulting in non transparency and non trackability [9]. Accountability might be providing a way for privacy protection within cloud computing by ensuring user trust and data transparency [10].

As the first step towards a comprehensive approach for data protection in the cloud, the authors have addressed the newly emerging data privacy problems caused due to data indexing in the case of cloud environments [11].

The concept of distributed accountability and innovative approaches for retaining data transparency in the cloud has been promoted towards automatic logging to the cloud, for data along with an auditing mechanism with the novel usage of JAR files [1].

As the continuing work the authors have discussed a cloud information accountability framework which concentrates mainly on the distributed accountability which is generally nonexistent in the cloud based environment. The authors argue for this kind of framework as necessary because of the fact that the outsourcing is completely under the control of Cloud Service provider and also that the client entities join the cloud and leave the cloud in a highly dynamic fashion and in random. This will enable ensuring accountability as a way of enhancement of trust and security in cloud based services [2].

When a password is used for the derivation of the key for the encryption process, the concept of Password Based Encryption (PBE) for securing data confidentiality emerges, which avoids the disadvantages of using public key cryptography [12].The core concept and various aspects of PBE by a comprehensive approach along with the strong recommendations for its implementation has been det.ailed[13].

The usage of the digital signature for authenticating the source of message and the various message security aspects obtained from message digest algorithms especially commonly used MD5 has been described stating its various applications[14].The HMAC algorithm which is used for verifying the integrity of data passed between the between applications in a vulnerable environment has been described[14].

## III. Problem Domain

Within SPI the public domain is the problem domain. In cloud based services, entire process is managed by using a CSP( Cloud Service Provider) who provides the cloud service and data availability. It is the Cloud provider who incorporates the computing infrastructure, runs the cloud software and delivers services to consumers through network access. Even though the flexibility of computing is very much elevated with all its benefits, there can be various security and privacy issues with respect to the shared data.

Making Cloud accountable includes provision of data transparency and trackability[10].The accountability introduces the trust relationship between various entities of cloud computing scenario and also helps to keep up the developer's authority over data in transit and in sharing. Security problems are mainly associated with the cloud for managing a dynamic environment of service. The challenge is to ensure that only authorized and privileged entities can gain access to it. So cloud providers must establish a trust relationship among their customers by providing transparency and accessibility. It is critical to have appropriate mechanisms to prevent cloud providers from using customer's data in a way that has not been agreed upon[9] .Also it is very important that the access control can be easily managed and well administered.

## IV. Problem Identification

Lack of user control can increase privacy issues .Thus the accountability feature becomes an important factor for providing a privacy scenario. This is mainly because processing power is handed over to a third party called the CSP who will have entire control over the data. A "Cloud Information Accountability Framework" (Distributed Accountability Framework) can be considered that can make data transparent and trackable[2]. The framework can provide the distributed accountability feature along with logging and auditing facilities for cloud environments by exploiting the advantages of JAR files. By using JAR files the problem of data loss can be addressed.

JAR files are considered for data sharing in the cloud since it's a compressed file format and required less storage space .The portability feature of the JAR file is also a reason for the selection. The complex structure of JAR files in the nested form is considered [2]. But there is no provision of Protection of JAR files from unauthorized access. In other words the security aspects are not given higher priority since





DOI: 01.IJNS.5.1.5

the stress is on the accountability [2]. The advantages of using the JAR files cannot be exploited if it cannot be properly authenticated.

Data security issues are discussed by Deyan Chen and Hong[5].The lack of auditing facilities and improper authentication and authorization procedures are matter of great concern in the case of cloud. Strategies for data protection and access control are also a necessary requirement in this aspect.

In the related works the accountability is given more importance and priority than the integrity of data and its content[2]. Integrity of shared data is important especially when the processing is happening in a completely different environment from those of actual data owner's environment, along with the accountability.

Even though the prime concern of data integrity lies with the owner, only the CSP can provide the integrity assurance. Hence the question of the owner downloading the data for verification, editing the data for correction and further uploading are the issues which are not practical since the cloud is highly dynamic. Therefore the integrity of data must be ensured at the CSP level itself.

Malicious clients which will arouse a problem for data privacy by impersonating the legitimate users, can be prevented with client side security measures. The client side security is ensured by the use of key verification process for each data access by the client. In addition to distributed accountability, the PBE system can keep log files tamper proof.

## V. Problem Statement

Suppose a user A wants to place a particular data on cloud for utilizing the resources in cloud that user may have the following requirements.

- Data must be only used by consumers who have the access privilege provided by the data owner and not by the CSP.
- Log files should be periodically sent back to data owners to inform current usage of their data
- Log files should be kept tamper proof.
- In any case of disputes the corresponding user can have access details of a client.
- Requires Auditing Facilities based on Log records.
- As a further extension there should also be strategies for data protection and access control including authentication and authorization.

## VI. Proposed System

In the proposed system an enhanced client information accountability framework is described [2]. The framework envisages a working scenario with the participation of the CSP, Client and Data Owner as well as the implementation of accountability and security on data in the cloud.

### *A. The Strategy*

- Data owner should be assured of the integrity of data in the cloud by the prevention of a client outsourcing the same data from the cloud.
- Client side security is to be ensured by the use of an algorithm which should have the properties such as:

i. Unique identifiability based on every different file even by one bit

ii. Repeatability of the above unique identity for the same input ID of the client.

iii. Irreversibility property should be satisfied that the unique identity does not lead to the file of origin in the cloud.

- The security of JAR files is included as a necessity to avoid unauthorized access and related issues. The authentication of JAR files has to be done by a valid signature.

In the case of cloud, conventional centralized access control policies will not permit the accountability feature, since the data owner will not have control over the data, its processing and storage. To enable such a control, several flexible logging and auditing facilities are required to be implemented. Besides, the logged data must be preserved unaltered. The availability of log files is of another consideration. In addition to that there must be well defined round robin auditing facilities involving CSP, Data owner and Client. Security features are also given higher consideration for better services from the cloud.

The Cloud information accountability framework is a system where the mechanism of JAR file in ensuring accountability by logging and auditing is discussed. The log of data access through JAR files provides wide range of information such as ID, Action, Location, and Time of access and checksum based on hash function [2].

But in the present work a subset of factors ID, access right for read or write (modify/edit a file) are considered. The Location and time of access factors [2] are of lesser priority from the point of view of implementation of security. The Checksum [2], for the previous history of access, is avoided because in the proposed system the log records are sent immediately to the owner following the current access as the focuses is on the current usage of data.

This framework fulfills confidentiality, integrity, availability and accountability requirements of a secured framework through JAR file authentication, MD5 based Client side verification and content integrity assurance by HMAC towards client outsourcing in to the cloud. Securing the framework includes the following processes.

a. The MD5 will cause to generate the unique identifiability in the form of message digest of a particular message or data. Thus keys generated by MD5 is used for providing an additional client side verification for each client access in order to enhance security of data.

b. The DSA algorithm is used for the authentication of JAR files by using digital signatures, to ensure the advantages of using the JAR files.

c. The HMAC algorithm provides the integrity of data, which is one of the major security aspects in the proposed framework.

d. The PBE (Password Based Encryption) scheme is used for log file protection[12].




DOI: 01.IJNS.5.1.5

*B. Algorithm*

- A cloud model is generated with the components CSP, Data Owner and Client.
- Enhanced Accountability Framework is ensured with the client as the third participant.
- The Data owner has to register with the CSP, which is authenticated using certain service level agreements (make it trusted) followed by the generation of OwnerID, by which it can subsequently upload the data file for global client access. This phase is categorized by the generation of a separate file containing owner ID, by the CSP.
- The framework will allow only privileged clients. The Client has to initially register with CSP with the details such as data filename, the required access privileges, client name and emailID in the registration form where a randomly generated Client ID along with a key is obtained which are supposed to be remembered by the client for later use at the time of LOGIN for data file request.
- A copy of the registration id and the access privilege is sent to the data owner automatically. The registration phase is over.
- Now the data owner can login after signature generation and verification based on DSA algorithm on the previously saved file containing owner ID.Following it, the owner registration table is updated to include the signature.
- Then it can create JAR files with the contents like access policies, the data item, HMAC code and access privileges, owner ID and Client ID. It is sent to CSP.
- Authentication of JAR files is done by checking whether it is containing the ownerID which contains a valid signature, before it is accepted by the CSP towards enhanced security.
- The JAR files contents are retrieved by the CSP .The client ID, access privilege along with HMAC code corresponding to the requested file are saved separately in a table which can be used for integrity verification.
- The client searches and identifies a file of interest from the cloud. The login phase into CSP begins for file access. The key verification phase using keys generated by MD5 is added for extra client side security which will be followed by the data file request by the client.
- The Log file for current access is generated at CSP.
- The Log file is protected by the "password based encryption" before the file is sent to the owner.
- The Owner permission is based on the log files and access privilege verification.
- Auditing is to be done based on this log files. Push based and pull based auditing facilities are performed.
- The owner responds after the auditing phase by the Permission "YES" or "NO" to the CSP.
- CSP will provide the data to the client based on owner permission.
- There is the possibility that the same data may be outsourced by the client into the cloud .This can result in the duplicity of critical data(atomic, military, intelligence, personal etc) owned by the owner, but in the name of the client. Such an activity is automatically prevented by the HMAC factor. The integrity of the data can thus said to be not compromised.
- Integrity verification is done for outsourced data by comparing the HMAC codes of outsourced data and the data in CSP corresponding to the client ID.
- Outsourcing is successful only if HMAC codes differ where the client becomes the owner.

*C. Framework Overview*

In this paper the notion of accountability is used for assuring the data governance by managing usability, security, integrity and authenticity of data shared between various clients. Hence the accountability aspect is having different phases. In the first phase the actions which must be logged are decided .Here it will be in the sense of creation of JAR Files and setting access privileges. The next phase is data processing and accessing. This is the phase where the entire process begins. The next process is the logging of necessary actions with accountability. These log files will be used as evidence in case of disputes or any other problems. The next step is to keep the Log files securely to protect it against any disclosure or security breaches. Some strong cryptographic methods can be used for this purpose.

Next phase is the reporting on the basis of the created Log file which contains the user access information. Based on the log files, distributed auditing can be done for provisioning a decentralized framework. Auditing is the process to ensure that CSP's are doing their work perfectly meeting all performance and security compliances. Finally based on auditing, problem areas are identified.

The next important function of proposed framework is to create a secure environment for efficient data sharing. For that purpose some of the disadvantages of the existing framework like lack of data integrity, lack of client side security and so on are identified and made a much better framework which is highly reliable and flexible[2].

*D. Enhanced Accountability Framework Schematic*

The Client Information Accountability framework that corresponds to an Enhanced Accountability framework, which is mentioned in the proposed system, to solve the above described problems is discussed here, which is shown in Fig.2. This is a secured client information accountability framework that can provide security and accountability features along with transparency and trust. There are mainly three loosely coupled components. Those include:

- Data Owner-who actually owns or develops data
- CSP-who is responsible for data provisioning to clients
- Client-who consumes data or services and will be receiving the access privileges

Accountability feature provided by this framework can be used to regain the lack of trust which customers have on cloud service provider. The main application is that it can verify the integrity of outsourced data. This will also provide additional client side security and enables protection of log files. All aspects of security like confidentiality, integrity,authentication and accountability is being covered by it.





DOI: 01.IJNS.5.1.5

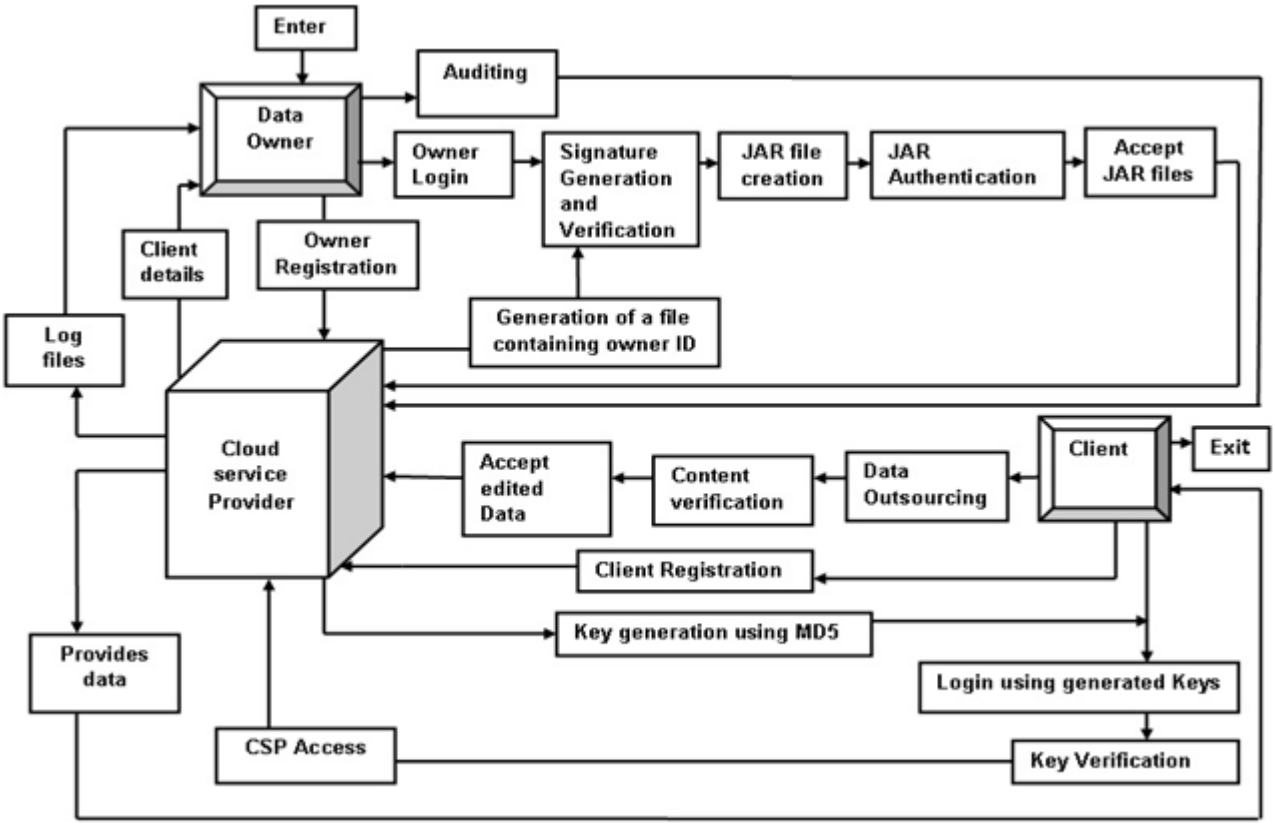


Figure 2. A broader schematic diagram of Enhanced Accountability Framework

*E. The Process*

The Enhanced Accountability framework provides the added security features. The entire processing is explained in the following steps.

Step 1: The processing begins with registration process as shown in the following Fig.3.Before that, CSP is assumed to be authenticated by using certain service level agreements, before the processing starts. Initially, data owner registers with CSP, with the file it wants to outsource. This is followed by the creation and storage of a file containing a copy of registered ownerID by the CSP.

Step 2**:** The client registration with CSP happens as the second part if client is searched out for some data from CSP and it is found there. The client registration proceeds with key generation which is used during client side verification.MD5 is chosen as the key generation algorithm.

Step3: The client details will be sent to the corresponding data owner. The details include client ID and access privilege. The data owner stores it for auditing process. Only registered data owners will exercise such a facility. Before that, the client details will be stored on the CSP for the further proceedings.

Step 4**:** The next step will be data owner login followed by signature generation and verification and the JAR file generation which is done based on the client details. Before that, the generated signature is used to update data owner's table. JAR files are characterized with a nested structure consisting of inner and outer JAR's. JAR files will be having access privileges for a particular Client ID. It will also contain the HMAC code of the data which is to be shared and also the

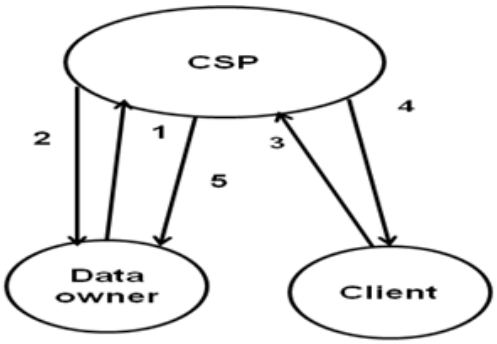


Figure 3. Registration Process

data itself along with client ID and ownerID.

*1) Authentication Algorithm:* In step 5, the JAR Authentication process is performed, before the JAR files are transferred or it is gets accepted by the CSP as detailed in Fig.4. The JAR authentication is done in the proposed system as a major enhancement. The authentication process is helping to protect the JAR files from copying attack, where some unauthorized entity will be trying to copy contents of JAR file and disassembling attack(Changing order of the details included) by malicious entities.

In the case if authentication is failed the transfer of files will be blocked and hence the JAR files are rejected by CSP.





DOI: 01.IJNS.5.1.5

**/*Signature Verification and JAR file transfer*/**

**Step1:**Data owner submit registration form
**Step2:**Generation of a file containing the registered owner ID in CSP
**Step3**:For login, enter OwnerID
**Step4**:If (signature generation and verification is success)
{ "update owner table"
"login success"}
else { "login failure"}
**Step 5:** After login, create JAR files
**Step 6**: If (JAR file is associated with OwnerID having a valid signature?**)**
{ " Authentication success"
"file transfer succeeds"
}
else {"Authentication Failure"
"file transfer fails"
}

Figure 4. Process of authenticating JARs

If the authentication succeeds then the JAR file will be received at CSP. Then the inner and outer JAR files are separated. Data processing will be done and corresponding client ID, the HMAC code and its access details, retrieved from JAR files are saved separately at CSP for integrity verification during outsourcing.

*2) Key Verification Algorithm:* In step 6 there is Client login followed by making Data request. This is the part where client side key verification happens as shown in the Fig.5. This verification can make client side secure and ensure efficient working.

Step 7: Then next step is the Log file generation by the CSP, based on data request. This is the way of ensuring current usage of data trackable. According to the framework each and every client access is logged .These log files are used for auditing purposes and to verify the integrity of data by the data owner. The log file must contain useful but only relevant information otherwise the time to create different log files will be high, which can introduce unnecessary overhead. Usually log files are encrypted and sent to the Data owner for keeping it secure.PBE is selected for encryption considering less memory requirements and more acceptable file numbers.

**/*Client side Verification*/**

**Step1**:Client submit registration form
**Step 2**:Generation of Shared Key(say k) for the client using MD5
**Step3**:for login, client enter Client ID and k
**Step4**:If (k==Key corresponding to client ID in CSP)
{ "Login success"}
else {"Login Unsuccessful"}

Figure 5. verification process done for client side

*3) Auditing Algorithm:* In step 8, based on the log files, auditing happens, as given in Fig.6. If the verification fails, that particular client Id will be deleted from owner side for blocking the client from future access. There are 2 types of auditing.

**/*Auditing*/**

**Step1**:Log files received at owner
**Step2**:If (access privilege of client in log file)= = (access privilege corresponding to client ID in Owner)
{ "Owner permission ="Yes"}
else {"Owner permission ="No"}

Figure 6. Process of Auditing

- Push based—periodic for each and every request
- Pull based—on demand approach

Step 9**:** Owner permission will be given based on the auditing. Actually through auditing the data owner is comparing the access privileges. If the access privileges are same then only the permission is "YES" otherwise it will be a "NO". The entire process is shown in above Fig.6.

Step 10: Client access is possible only if Data owner's permission is a "YES". If the permission is granted by data owner then the CSP will immediately provide the data to the required client. A Client has two options:

- View-read access privilege
- Download-write access privilege

Step 11**:** The next process will the client trying to outsource the data giving it client ID and uploading data in to the CSP, which is to be outsourced.

*4) Integrity Verification Algorithm:* In step 12**,** Integrity verification for outsourced data is happening. This is done by comparing HMAC codes as detailed in Fig.7. As the client is uploading data with its own ID, the uploaded data's HMAC code will be compared with the HMAC code of data corresponding to Client ID in the CSP. If it matches outsource will be a failure. Thus integrity of data is also maintained.

**/*Integrity verification*/**

**Step1**:Client enters Client ID and data(D) to be outsourced
**Step 2**:Generation of HMAC (D)
**Step3**:If (HMAC(D)= =HMAC(data for corresponding ClientID in CSP)
{ "unauthorized Process"
}
else {"authorized user and integrity verified start outsource the file"
}

Figure 7. Process of verifying integrity of data

The proposed model allows different types of data such as image files and text files would improve the flexibility of the framework but the storage overhead caused by it cannot be ignored.




DOI: 01.IJNS.5.1.5

*F. Functions*

The main functions of the accountability framework can be summarized as the following.

*1) Authorizing Cloud Users:* Data owner must have to register with the CSP to be an authorized customer. This can be considered as a preliminary step towards JAR file authentication. Client registration to CSP is also there to ensure client side security. Client registration proceeds with the generation of the key using MD5 which can be used for client side verification [6].Then Client registration details will be forwarded to the data owner from CSP.

*2) JAR File generation:* Here the Data owner stores access privileges, services or data to be shared in the form of JAR files. In this work there is much flexibility in the selection of data files. All types of files like image file, text file etc can be included, improving the flexibility of framework. Access privileges (read, write etc) indicate which users or stakeholders can access it. In addition to that this JAR files also contain the HMAC code of the corresponding Data item which will be the basis of data integrity verification.

JAR (Java archive) file format is chosen because of its numerous advantages which include less storage space since it is a compressed file format .To an extent the JAR can prevent data loss that can possibly occur. Here nested JAR files are used .Usually it can have numerous files in a single file. Outer JAR covers one or more inner JARs. Outer JAR contains authentication details and inner JAR contains access details and data for the client. The complicated structure can add additional protection to all data items in it.

*3) JAR File Authentication,CSP Authentication And CSP Storage:* JAR authentication is an essential step involved in order avoid to anonymous component access while transferring to CSP for storage. JAR Authentication can be done by using Digital Signature generation and verification for corresponding owner[7], which is shown in Fig.8.

a) Digital signature based on DSA algorithm

The sender uses a signing algorithm to sign the message. The receiver receives the message and the signature and verifies it. If the result is true, the message is accepted; otherwise, it is rejected. Digital signatures validate the authenticity of JAR files and acts as a means of non repudiation [7].

➢ Digital signature DSA Signature Creation and Verification

For generating digital signature the following procedure has been implemented using NetBeans IDE. A digital signature is created or verified using an instance of the Signature class. The signature is generated and verified with login process and involves the following steps.

- Select the public key, private key and a random value or a secret number per message and the compute the signature pair.
- Generate the hash code out of the data(file containing ownerID) using SHA (Secure Hash Algorithm).
- Generate signature based on the hash function along with a random number generated for this particular signature and the private key.
- Verification involves the comparison of the signature components along with the generated hash code of the data. If the matching succeeds then the verification is considered to be successful[14].

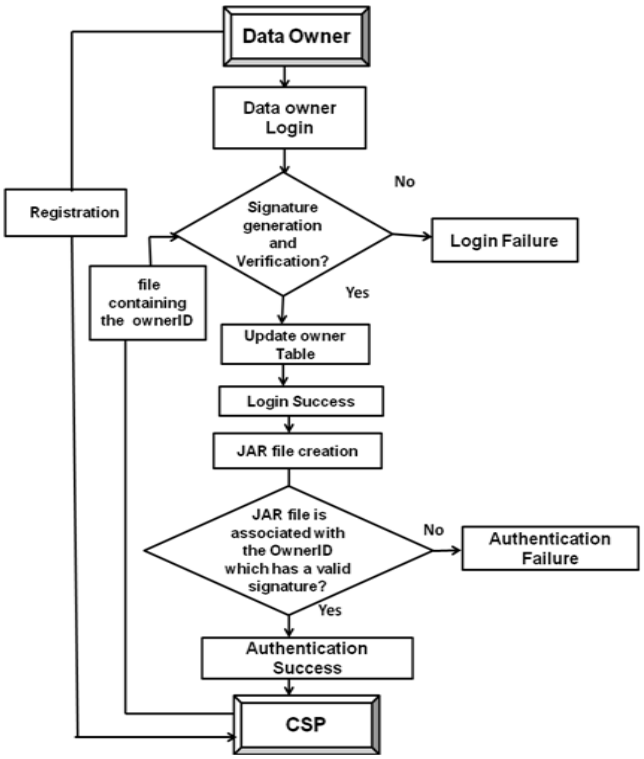


Figure 8. Signature verification and JAR Authentication

This entire processing of Signature generation and verification will be done at the time of data owner login. So before the JAR files are accepted by CSP, an authentication step is carried out on the basis of Signature ID's generated using DSA algorithm for the corresponding owner ID. If this signature ID generation and verification by the CSP for the corresponding user succeeds, then only JAR files are considered as authenticated and received by CSP.

This can avoid attacks like disassembling of contents for changing order or completely destroying it and copying attack. Also before storing in CSP, CSP must be authenticated with respect to certain level agreements for making it a trusted entity.

*4) Log Files Generation To Data Owner:* Each and every access will be logged .Usually log contains client accessibility details. Actually it will be having a list of actions occurred during entire data sharing with respective to the framework. A log record takes the following form

$$r_i = [ID, Data, Act]. \quad (1)$$

where $r_i$ represents a log record for an entity (here identified by $ID$ with corresponding action $Act$ on the data item $Data$ . Also each log record will be in an encrypted form, in order to keep it tamperproof. For that purpose PBE algorithm can be used[12].

*a) PBE Algorithm Activity*

PBE is a combination of hashing and symmetric key encryption. This will generate a secret key based on a





DOI: 01.IJNS.5.1.5

password. There is no problem of securing storage of private key in the case of PBE[12].Moreover it can provide the confidentiality features also that are necessary and it is implemented through NetBeans IDE.

A random value called salt is mixed with the password to avoid pre-computation attacks .Usually it is generated by using a random number generator and will be written to the encrypted file for decryption. Key derivation can be made complicated by using another value called iteration count which is the count of number of times the hashing applied to the password and salt combination[12].

The start phase of the process is whenever a client is making a data request after login. In this work PBE with MD5 and DES are considered for log file encryption[13]. The process is discussed below

➢ *Encryption Operation*

1. Input the password
2. Select salt value say S and iteration count c
3. Apply MD5 to password P, the salt S, and the process is repeated c times.

$$DK = MD5(P, S, c, DkLen). \quad (2)$$

p-password
s-salt
c-iteration count
DkLen-length of derived key
DK-derived key

4. Encrypt the message M(log file) with DES under the derived key DK to produce a cipher text C.

This encrypted log file is sent to data owner by the CSP and there the decryption process is done.

➢ *Decryption Operation*

1. Obtain the salt S for the operation.
2. Obtain the iteration count c for the key derivation function
3. Obtain the DKLen, for the derived key for the underlying encryption scheme
4. Apply the selected MD5 to the password P,the salt S, and the iteration count c to produce a derived key, DK of Length, DKLen octets:

$$DK = MD5(P, S, c, DkLen). \quad (3)$$

5. Decrypt the cipher text C using DES under the derived key DK to recover a message M.
6. Output the recovered message M

*5) Auditing:* Auditing is another important phase of the Framework. It will be based on log files. It is done to check that the functioning is like what they are claiming to be, against actual facts in order to verify compliance. Usually here distributed auditing is done to get an end to end accountability. There are two types of auditing, which include:

• Push Mode –This is the case where the log files are periodically being sent back to data owner .It is done automatically with each client access.

• Pull Mode-This is an alternative approach in which log records are retrieved by Data owner as they needed. It is usually performed when there is problems regarding the security of data.

*6) Client Access:* Client must be registered with CSP and it must login with CSP to access the necessary data. Before accessing CSP it must be passed through a key verification step with those keys provided generated using MD5 during registration[6]. The process of key verification is shown in the Fig.9.

*a) MD5 Algorithm Activity: T*his algorithm takes input of any arbitrary length (client ID) and produce output as 128 bit message digest[14]. The processing involves the various steps such as adding the Padding Bits, Appending 64 bit representation of original message, Initialize the MD Buffers, Processing Message in 16-Word blocks and the final output is the 128 bit message digest. The MD5 algorithm activity has been implemented by NetBeans IDE in the work.

The final message digest which is considered as the key obtained will be in the form of byte array (16 bytes).So it will be converted to hexadecimal to make it readable during processing .

Thus it will be providing an additional client side authentication. This verification process can avoid a lot of complexities and security issues that can occur from client side. Based on the access privileges client can have two options .They are

• View-In the case of view reading of data is allowed. There is no option of saving any raw copy of it permanently. When there is a read request, temporary file is created which can be viewed at client side.

Download-In this case the entity is allowed to save a raw copy of data .Here a copy of decrypted file will be present at client's system. In the case of image files only download option is possible.

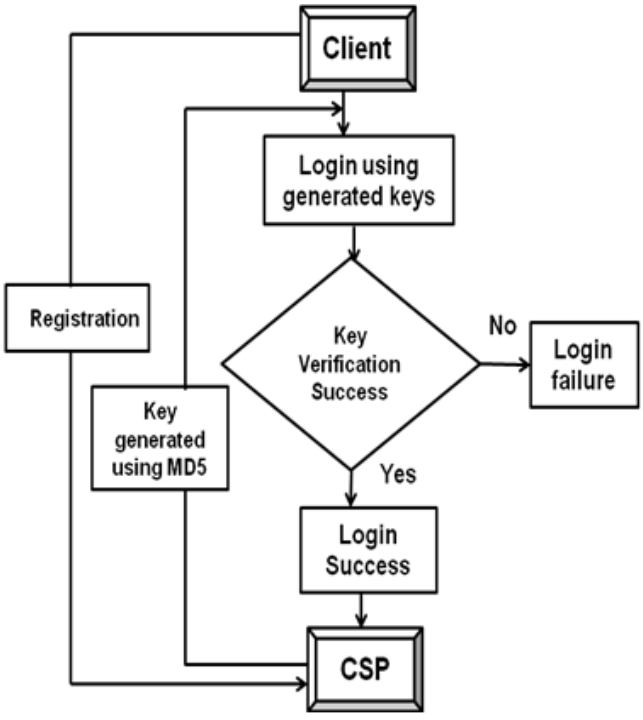


Figure 9. Client side verification

*7) Integrity Verification:* Finally the data owner can verify integrity of its data in the CSP on the basis of log files and




DOI: 01.IJNS.5.1.5

auditing. In addition to that CSP also checks for integrity of a particular data which is outsourced by the client which is shown in Fig.10.Each time when the outsourcing happens it will check uploaded data using HMAC algorithm[14].This is implemented using NetBeans IDE and involves the following features.

*a) HMAC Algorithm Activity:* HMAC is a keyed-hash message authentication code and is be used to verify data integrity of the data the client intend to outsource further. This is a special type of mac (message authentication code) which is a combination of hash function and key and is very fast and The message authentication code

$$MAC = C_K(M). \quad (4)$$

condenses a variable-length message M(fileuploaded/files owned by data owner ) using a secret key K to a fixed-sized authenticator. HMAC is the hashed mac function which is described in various literatures[8]. It is defined as

$$HMAC(K,m) = ((K \oplus opad) \parallel H((K \oplus ipad)) \parallel m)) \quad (5)$$

where

- $H$ is a cryptographic hash function,
- $K$ is a secret key padded to the right with extra zeros to the input block size of the hash function, or the hash of the original key if it's longer than that block size.
- $m$ is the message to be authenticated
- $\parallel$ denotes concatenation
- $\oplus$ denotes exclusive or (XOR),
- $ipad$ = 00110110 (36 in hexadecimal)
- $opad$ = 01011100(5C in hexadecimal)

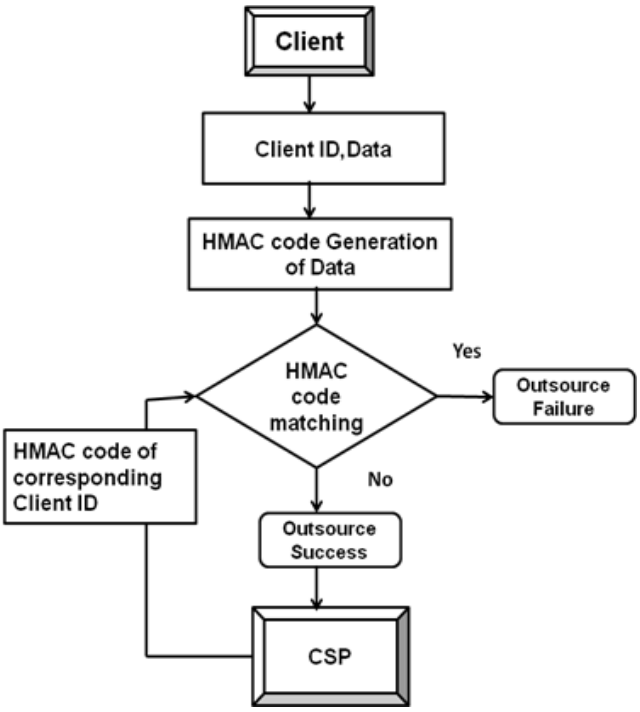


Figure 10. Integrity verification

For this initially when JAR file is created by data owner corresponding HMAC code of data is also included. When the HMAC code of uploaded content does not match with original code then it will be accepted. This comparison is possible since the CSP will always store the details like which client is provided which data along with its HMAC code, and when a particular client submits file a data along with its ID, the CSP can easily verify it .Indirectly it is the way of ensuring integrity by verifying if any client try to upload the same data which is registered under some other owners ID. In this way each and every client access can be made secured and accountable.

## VII. Experimental Set Up

The experimental set up is a virtual environment.

There are basically three virtual machines involved which have been set up using VMware workstation software for three components a) the Data owner b) the CSP and c)the Client.These virtual machines are communicating with each other to model the enhanced accountability framework discussed in the proposedsystem.

## VIII. Input-Output Model

An input-output model is provided in Table. I, where the input data is related to the output data through the processes. For easier understanding, the feedback loops are not shown.

## IX. Simulation

Java is used as the programming language for implementing the framework with MYSQL database as the backend. The Java.net package is useful for socket programming, thus providing the inter- communication between data owner, Client and the CSP. NetBeans is an Integrated Development Environment mainly aims to create an efficient software development environment providing the required tools. NetBeans IDE provides comprehensive support to most of the newest Java technologies. The functionalities of IDE are provided as modules. In the work, the modules such as authorization of cloud users, JAR file generation and authentication, Log file generation, Auditing and integrity verification have been implemented in the NetBeans IDE using various API's and MYSQL database. The VMware Workstation enables the users to set up multiple virtual machines (VM's) in order to be used along with the actual machines, thereby enabling the access of the virtual machines running anywhere in the system in windows or Linux environment. Thus a cloud environment has been set up using the VM's running across a single system with multiple OS's providing the corresponding responsibilities. Samples of input and output data along with its description is given in Table. II.

## X. Result Analysis

The implemented framework will be having the dataflow happening between the three main components Data owner,





DOI: 01.IJNS.5.1.5

TABLE I. INPUT-OUTPUT PROCESS TABLE

| Input | Process | Output |
|---|---|---|
| Name, Mail Id, Name of data item | Data owner Registration | Owner Id, Separate file containing OwnerID |
| Owner Id | Data owner Login | Signature generation and verification ,updating owner table |
| Name of data item | Searching | Item found/not found |
| Name, Mail Id, Name of requested item, privilege mode | Client Registration | Client ID, Key generated using MD5 |
| Client ID, privilege | Intimating Data owner about registered clients(sending client details) | Client ID and privilege reached at Owner side |
| Data Item | HMAC code generation | Corresponding HMAC code |
| Data item, Owner ID, HMAC code, Client ID, Privilege | JAR file generation | JAR files |
| JAR files | JAR Authentication | Authenticated JAR files |
| Authenticated JAR files | JAR file content extraction | Data item, Owner ID, HMAC code, Client ID, Privilege, (inner JAR and outer JAR separated) |
| HMAC code, Client ID, Privilege | Process Data | Separate file is there with contents ClientID, Privilege and Hmac code |
| Client ID, Key generated using MD5 | Key verification | Authenticated client |
| Client ID, Data Item, Privilege Mode | Data request | Data item request forwarded to CSP |
| Data Request | Log file generation | Log file (ID, Access privilege for corresponding file) |
| Data item, Owner ID | Log file Encryption | Encrypted Log file |
| Encrypted Log file | Log file Decryption | Decrypted Log file |
| Decrypted Log file Contents | Auditing | Owner permission |
| Owner permission(“Yes”) | Client Accessing | View/Download |
| Client ID, Data file | Outsourcing | HMAC code generation and integrity verification |
| HMAC code of uploaded file, HMAC code of file corresponding to client ID | Integrity verification | CSP downloads and stores file otherwise rejected |

CSP and Client which are in three different VM’s. The data owner is indirectly related to the client as the client access is authenticated by the data owner. Thus client is included as an important component in this new framework. The result analysis is shown in Table. III, as activity table. The comparison of results mainly concerns with the activities as narrated in Table. IV.

## XI. CONCLUSION AND FUTURE WORK

In the related works[1][2] even though the accountability feature is considered ,the security aspects are given lesser priority. In this paper an enhanced security framework is being described that can provide the accountability feature which is one of the most important requirement for cloud based services along with added security features like content integrity verification and JAR authentication. This framework can be a solution for many of security and privacy related issues existing in a cloud computing scenario.

Simple techniques such as DSA based digital signature, HMAC, MD5 and PBE are used for security enhancement to data access by a client in the cloud. This framework can assure a secured and accountable data sharing and can be used to improvise the deployment of cloud based services in a highly trusted environment

The outsourcing by the client implies a situation where the client attempts to modify the original data, with change in HMAC value, which will result in the acceptance of the data in the name of client only by the CSP but not in the name of the data owner who has provided the data to the client, thereby ensuring the integrity of original data.

The framework and coding has to be enhanced for the




DOI: 01.IJNS.5.1.5

*A Typical Sample Data RUN*

TABLE II. TYPICAL SAMPLE DATA

| Description | Sample data-Input/output data | Remarks |
|---|---|---|
| 1. Input : Image/Text data → |  or good morning | |
| 2. HMAC code of a file after base64 encoding → | OLMiEfTnAI161412NVNH7Q== | |
| 3. Contents of Log file after Encryption → | HXºw‰„¦™Å‡ZšbRù4¤GG0ÕI¡ßª ™ÛÂ | |
| 4. Contents of Log file after Decryption → | 243,sa.txt,Write | |
| 5. Signature ID generated using DSA → | [B@17bd6a144 | |
| 6. Key generated using MD5 → | c4ca4238a0b923820dcc509a6f75849b | |
| 7. Downloaded Data from cloud by the client → |  or good morning → | After Owner permission |
| 8. Outsourced / uploading of data by permission → |  or GOOD morning → | Permitted because the outsourced data is different from the original downloaded data of the data owner |

access of video files, since a video provides a stream of images unlike the processing and sharing of a single image file , single text file , pdf file etc. The manual process of owner interaction with the system to permit the client to access the cloud may be automated for faster processing. In the present work the HMAC ensures the existence of files of the owner as unique. No duplicity of the same file is permitted. Hence the right to edit the unique files by the clients may be introduced overriding the HMAC feature.

## ACKNOWLEDGEMENT

The authors wish to thank the CSE department for their support and help in completing this work.

## REFERENCES

[1] S.Sundareswaran, A. Squicciarini, D. Lin, and S. Huang,"Promoting Distributed Accountability in the Cloud," Proc. IEEE Int'l Conf. Cloud Computing, 2011.

[2] Sundareswaran, Smitha:Squicciarini,Anna Cinzia:Lin,Dan "Ensuring Distributed Accountability for data sharing in the cloud" IEEE Transactions on Dependable and Secure Computing,2012

[3] Bhushan Lal Sahu,Rajesh Tiwari, "A Comprehensive Study on Cloud Computing" International Journal of Advanced Research in Computer Science and Software Engineering ISSN:2277 128X Volume2,Issue 9,September 2012

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

[5] Deyan Chen,Hong, "Data Security and Privacy Protection Issues in Cloud Computing",2012,International Conference on ComputerScience and Electronics Engineering

[6] Performance Analysis of MD5 Joseph D. TouchUSC / Information Sciences Institute(touch@isi.edu) Appeared in the proceedings of Sigcomm '95, Boston MA.

[7] Sreekanth Anyapu, G. Aparna, R. Manognya, D. Ravi Kumar" Message Security Through Digital Signature Generation and Message Digest Algorithm" International Journal of Emerging Technology and Advanced Engineering Volume 3, Issue 3, March 2013





DOI: 01.IJNS.5.1.5

TABLE III. RESULT ACTIVITY TABLE

| Activity | Client | Data Owner | CSP | Remarks |
|---|---|---|---|---|
| Authentication | | | Yes | Based on certain service level agreements |
| Registration | | Yes | | CSP generates a file with a copy of Owner ID |
| Search | Yes | | | |
| Registration | Yes | | | |
| Key generation | | | Yes | |
| Sending client details | | | Yes | For data owner |
| Owner Login | | Yes | | |
| SignatureID generation and verification | | | Yes | Update the owner table to include the signature |
| If TRUE/FALSE | | Yes/No | | Login Success-Yes/No |
| JAR Authentication | | | Yes | Of generated JAR files |
| Owner ID verification for a valid Signature | | | Yes | |
| If TRUE/FALSE | | Yes/No | | JAR file access by CSP-Yes/No |
| JAR file Transfer | | Yes | | |
| JAR file reception | | | Yes | |
| Extraction of JAR file contents | | | Yes | Needed for integrity verification and other purposes |
| Client Login | Yes | | | |
| Key verification | Yes | | Yes | For ensuring client side security |
| Data request | Yes | | | |
| Log file generation | | | Yes | |
| Log file Encryption | | | Yes | |
| Encrypted Log file Transfer | | | Yes | |
| Encrypted Log file Reception | | Yes | | |
| Decryption of log files | | Yes | | |
| Auditing(push/pull) | | Yes | | |
| Owner Permission | | Yes | | |
| IF YES/NO | Yes/No | | | Client access-Yes/No |
| Permission verification | | | Yes | |
| Client Access<br>Client View | Yes | | | |
| Client Download | Yes | | | |
| Client Outsourcing of downloaded file | No | | | Not permitted |
| Submitting of Client ID and Downloaded file | Yes | | | To CSP |
| HMAC Code generation | | | Yes | Of Submitted file |
| HMAC Code Matching | | | Yes | Of submitted file and those in the CSP corresponding to Client ID |
| IF TRUE/FALSE | No/Yes | | | Outsourcing of submitted file denied-Y/N |
| QUIT | Yes | | | |




DOI: 01.IJNS.5.1.5

TABLE IV. COMPARISON

| Proposed System | Ref[1][2] |
|---|---|
| An Enhanced Accountability framework known as Client Accountability Framework | Accountability Framework is not having enhanced features for cloud security |
| Client Side Key Verification | Nil |
| JAR Authentication | JAR files are used but not Authenticated |
| PBE based Log file protection | Nil |
| Preserving data integrity | Integrity of data is not given much importance |
| Secured Outsourcing | Nil |
| Involvement of different types of files like text files and image files | Only image files are considered |
| Feedback mechanism for owner permission | There is no such mechanism for owner permission |

[8] Mihir Bellare ,Ran Canetti,Hogo Krawczyk, "Message Authentication using Hash functions:The HMAC construction",Appears in RSA Laboratories CryptoBytes Vol.2,N0.1,Spring 1996

[9] Takabi, Hasan:Joshi,James B D;Ahn,Gail-Joon, "Security and Privacy Challenges in Cloud Computing Environments",Security & Privacy,IEEE,2010

[10] S. Pearson and A. Charlesworth, "Accountability as a Way Forward for Privacy Protection in the Cloud," Proc. First Int'lConf. Cloud Computing, 2009.

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

[12] PasswordBased Encryption http://www.cs.ship.edu/~cdgira / courses/.../Article3-PBE.pdf

[13] PKCS# 5: Password-based cryptography specification version 2.0 http://tools.ietf.org/pdf/rfc2898.pdf

[14] William Stallings, "Cryptography and network security-principles and practice", Pearson Prentice Hall, 3rd Edition, 2002.

## BIBILOGRAPHY

Miss. Prema Mani completed her B.Tech from M.G University College of Engineering, India in 2009.She did her Post Graduation (M.Tech) in Computer Science and Engineering from Ilahia College of Engineering and Technology under the M.G. University, Kerala, India. Her area of interests are cloud computing and security

Mrs. Theresa Jose is an Assistant Professor in the Computer Science and Engineering Department of Ilahia College of Engineering and Technology, Kerala, India. She did her B.Tech in 2006 from St.Joseph College of Engineering and Technology, Kerala, India under the M.G University, followed by her Post Graduation(M.Tech,) at Viswajyothi College of Engineering and Technology, Kerala in 2009.Her research areas are data mining ,image processing and cloud computing.

Professor Dr.Janahanlal Stephen is the Research Dean in the Computer Science and Engineering Department of Ilahia College of Engineering and Technology, Kerala, India. He took his Ph.D from Indian Institute of Technology (IIT),Chennai, India.His research interests are in the area of system dynamic simulation by Prof.J.W.Forrester(formerly of MIT, USA), cloud computing, image processing, and security.




DOI: 01.IJNS.5.1.5