Identity Based Secure Distributed Data Storage Scheme using Nymble Server

Iris Elvy Gonsalvez I
PG scholar in CSE
Younus College of Engineering and Technology,
Vadakkevila, Kollam- 691010

Prof. Nijil Raj
Associate professor, CSE&IT
Younus College of Engineering and Technology,
Vadakkevila, Kollam-691010

Abstract: **In cloud computing data owner can outsource his data on cloud server and only authorized user can access or request for data. This service is called Database-As- A-Service. But there are different issues related to confidentiality, integrity and security of data. So we consider this as a basic idea for propose concept. The existing system does not provide security against possible attack of clouds, DDOS attack and misbehavior of user. But the propose system can provide security against collusion attack, DDOS attack and prevents misbehaving user. By using the concept of re-encryption the data get more secure and access permission that who will access the data is decided by the data owner. The file owner can decide the access permission independently without the help of the private key generator. Propose system uses wide possibility of mobile devices. In existing system, to provide better security data owner has to be online all the time. This is not possible always. Our propose system will eliminate this drawback by sending notification to data owner.**

1.INTRODUCTION

In Cloud computing technology includes many technologies such as the autonomic computing virtualization, utility computing, service oriented architecture and many. The purpose of these technologies to provide scalable, shared resources software and hardware and providing services over the network. The cloud term 'as a service' is referred to as providing something as a service over the network. There are 3 types of services provided by cloud as: Software as a Service (SaaS), Platform as a Service (Paas), Infrastructure as a Service (IaaS) and many more. All the provided services are based on policy of on-demand fashion in which users can pay only to for their required usage. Today, many cloud service providers such as Amazon's EC2 and S3, Microsoft's Windows Azure Google's App Engine providing the facility to different users. Users who cannot afford such huge cost to build their own huge infrastructure, so they can have their work done by the help of cloud providers at minimum cost. As per the type of users and the hosting of environment the cloud architecture can be divided as four types: 1.Public Cloud, 2.Private Cloud, 3.Hybrid Cloud, 4.Community Cloud. Public cloud provides services, are hosted for public usage and anyone can have their data stored and services get done using this cloud. Data security is most important issue here. Private cloud, where the data access and service usage are restricted to one authority only. In Hybrid cloud, it shared by a limited no. of organizations and it combined the features of both private and public cloud. Community cloud is much like the private cloud but in this the data is shared among the same entities of the one organization. Including this cloud also provides service called as database-as-a-service, in this data owner can outsourced his data/files on cloud server for reducing space cost as well as maintenance cost and only the authorized legitimate user can query /request this data. In this before uploading data on cloud server data owner has to be encrypt his data. Proxy servers can perform some functions on the stored cipher text without knowing anything about the original data/files. Cloud server is untrusted server because it managed by is untrusted third party. Therefore, here issues of confidentiality comes in existence ,data owner mostly concerned on the security from unauthorized access, integrity means correctness of the file/data after outsourcing to the proxy server, as well should not be modified by unauthorized user or even though by the proxy server. So, this is the reason that it becomes major research problem among research community and it growing day by day.

In this paper, we propose a system for security enhancement of cloud computing using identity -based encryption; it can capture the following properties:

(1) Without the help of the private key generator (PKG) the file owner can decide the access permission independently;

(2) Files are classified into easy accessible and sensitive file. Nymble server can give access to easy accessible file without permission of file owner. But in the case of sensitive file nimble server should send notification to data owner. For one request, a receiver can only access one file, instead of all files of the owner;

(3) Our schemes are secure against the DDOS, prevent misbehavior of user

(4) Also secure against collusion attacks, namely even if the receiver can compromise the proxy servers, he cannot obtain the owner's secret key.

(5) Can get the notification about user request on his mobile phone through short message service which is not possible in existing system.

II. RELATED WORKS

A proxy re-encryption is generally used when one party, say Bob, wants to reveal the contents of messages sent to him and encrypted with his public key to a third party, Chris, without revealing his private key to Chris. Bob does not want the proxy to be able to read the contents of his messages. Bob could designate a proxy to re-encrypt one of his messages that is to be sent to Chris. This generates a new key that Chris can use to decrypt the message. Now if Alice sends Chris a message that was encrypted under Bob's key, the proxy will alter the message, allowing Chris to decrypt it. This method allows for a number of applications such as e-mail forwarding law-enforcement monitoring, and content distribution Blaze, Bleumer and Strauss later presented the first secure "atomic" primitive: an Elgamal-based approach in which the proxy could not learn the message being processed. Unfortunately, the approach in is inherent bidirectional: a corrupted proxy can re-encrypt ciphertexts not only from Alice to Bob, but also from Bob to Alice. Even worse, collusion between the Proxy and "delegator" Alice could reveal the secret key of "delegatee" Bob. Jakobsson, and Zhou, Mars, Schneider and Redz partially addressed these concerns by proposing a quorum-based protocol which divided the proxy into many components. More recent works have focused on the development of unidirectional proxy re-encryption schemes, where collusion between a delegator and the proxy does not compromise the delegatee. Dodis and Ivan realized a form of unidirectional proxy encryption by using double-encryption (or by splitting a single decryption key into two parts). Their approach permits a form of single-delegation proxy re-encryption when parties hold pre-shared keys. Ateniese, Fu, Green and Hohenberger proposed an improved, non-interactive unidirectional scheme which removed the need for pre-shared keys and permitted arbitrary delegations.

A.Identity based proxy encryption

Identity -based proxy encryption (IBPE) was first proposed by Ivan and Dodis in which they presented security model for both unidirectional and bidirectional Identity Based Proxy Encryption schemes. In this scheme, PKG delegates decryption rights to all identities in the system. Such delegation is non divisible, i.e., PKG cannot delegate decryption rights for only a subset of identities in the system. This approach differs conceptually from non-interactive approach, where individual users delegate their decryption rights. In this system the master secret key, used to extract secret keys, split into two parts. One is sent to the proxy server and the other is sent to the user. The user can decrypt a cipher text for him with the help of the proxy server. Finally, the Dodis/Ivan system has significant security implications: collusion between the proxy server and delegate results in a system-wide compromise, allowing the colluders to reconstruct the Identity Based Encryption master secret.

B.Identity based proxy re- encryption

The first identity-based proxy re-encryption (IBPRE) was developed by Green and Ateniese where the proxy server can transfer a cipher text for the owner to a cipher text for the user after he gets a re-encryption key from the former. The IBPRE schemes divides into the following two types based on the generation of the re-encryption key:a) The re-encryption key can be computed by the owner .In this paper , to decrypt the cipher text, the owner selects a random number and computes a re-encryption key by randomizing his secret key. Then, he encrypts the selected random number under the user's identity. Finally, he sends the re-encryption key and the cipher text to the proxy server. Using the re-encryption key, the proxy server can transfer a cipher text for the owner to a cipher text for the user. The user decrypts the cipher text using his secret key and obtains the random number selected by the owner. Then, he can decrypt the re-encrypted cipher text by the random number. Unfortunately, these schemes are safe to the collusion attacks. If the user can compromise the proxy Server, they can decrypt the cipher text, obtain the random number selected by the owner and compute the secret key of the owner. b) The re-encryption key can be computed by the Private Key Generator. This system proposed by L. Wang, M. Mambo, and E. Okamoto, in this system, the PKG computes the re encryption key by checking the secret keys of the owner and the user.

Identity-based Secure Distributed Data Storage

This system, proposed by J.Han, Willy Susilo, and Yi Mu, a user's identity can be an arbitrary string and two parties can communicate with each other without checking the public key certificates. At first, the file owner encrypts his files under his identity prior to outsourcing them to servers. Then, he sends the cipher texts to the proxy servers. Consequently, the proxy servers can transfer a cipher text encrypted under the identity of the owner to a cipher text encrypted under the identity of the` receiver after they has obtained an access permission (re-encryption key) from the owner.

To provide confidentiality for the outsourced data, propose scheme provides the following properties:

1. Unidirectional: After receiving access permission that who will accessing which data, the proxy server can transfer a cipher text for Alice to a cipher text for Bob while he cannot transfer a cipher text for Bob to a cipher text for Alice.
2. Non-interactive: The access permission can be created by the file owner only without any trusted third party and interaction with him.
3. Key optimal: The size of the secret key of the receiver is constant and independent of the delegations which he accepts.
4. Collusion-safe: The secret key of the file owner is secure from malicious user even if the receiver can compromise the proxy server.
5. Non-transitive: Receiving the access permissions computed by A for B and B for C, the proxy server cannot transfer a cipher text for A to a cipher text for C.
6. File-based access: For one query/request, the receiver can only access only one file. This can improve the security of the outsourced files and is desirable to maintain the access record.

III.THE PROPOSED SYSTEM

In this paper, we propose a system for cloud computing where, for one query, the receiver can only access one file, instead of all files. In other words, access permission (re-encryption key) is bound not only to the identity of the receiver but also the file. The access permission can be decided by the data owner, instead of the trusted party (PKG). Here files are classified into two: (1) easy accessible (2) sensitive file. Here cloud server is named as nymble server. At first user and owner have to register with nymble server. Data owner and user should register by visiting nymble server site.There he should give his username,email-id,phonenumber and password which he wish to use.After checking the details the server authenticates the owner.The owner should use this username and password to login into server. After successful login he can upload his encrypted files. Data owner should keep secret key for encrypted sensitive data with him itself. When a user request for sensitive data, server should first send notification to owner. After checking user details, owner can decide whether to give permission or not. But he can authorize server to give encrypted easy accessible file to user without notification. After registering server gives a nymble ID for user. User should request for data with this nymble Id. When a user is detected to be misbehaving he is prevented from further access. Here blocking period is 48hrs from the time misbehaving is detected.Furthermore, our schemes are secure against the collusion attack and will also detect the distributed DOS attack which possible on proxy server in which server can't proceed to legitimate work and this system will available to owner on his/her mobile device. And can get notification also .

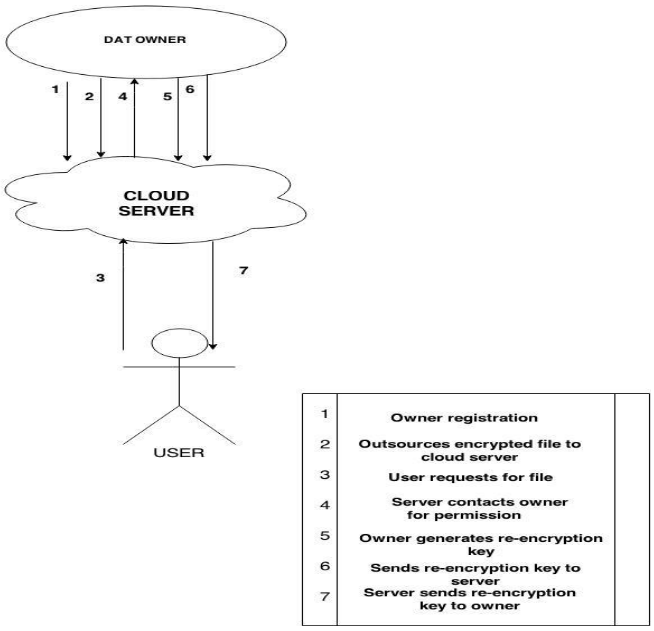

DAT OWNER
1
2
4
5
6
CLOUD SERVER
3
7
USER
1 Owner registration
2 Outsources encrypted file to cloud server
3 User requests for file
4 Server contacts owner for permission
5 Owner generates re-encryption key
6 Sends re-encryption key to server
7 Server sends re-encryption key to owner

Fig 1. Architecture

A.NYMBLE

After obtaining nymble id from the nymble server, the user connects to the nymble server. Nymble id is generated using users email address and phone number. When connection establishes with server nymble ticket is generated. Nymble tickets are bound to specific time periods. In our system time is divided into link ability windows of duration W, each of which is split into L time periods of duration T(i.e., W=L*T).We will refer to time periods and linkability windows chronologically as t1,t2….tL: and w1,w2….. respectively. While a user's access within a time period is tied to a single nymble ticket, the use of different nymble tickets across time periods grants the user anonymity between time periods. Smaller time periods provide users with higher rates of anonymous authentication, while longer time periods allow servers to rate-limit the number of misbehaviors from a particular user before he or she is blocked. For example, T could be set to 5 minutes, and W to 1 day(and thus L = 288).

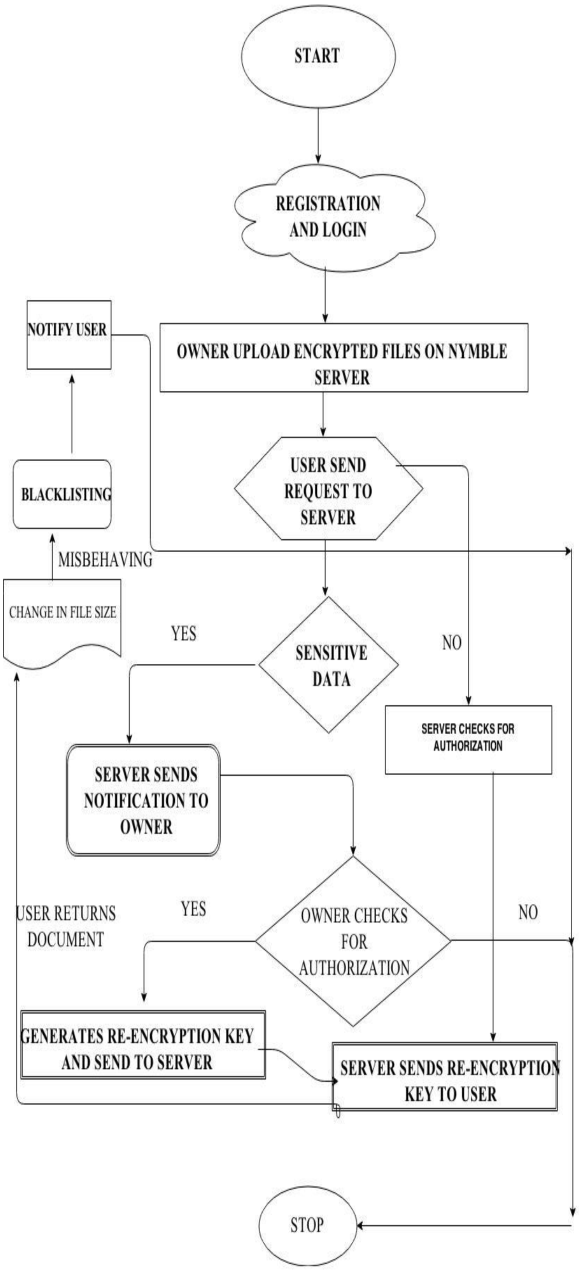

START
REGISTRATION AND LOGIN
NOTIFY USER
OWNER UPLOAD ENCRYPTED FILES ON NYMBLE SERVER
BLACKLISTING
USER SEND REQUEST TO SERVER
MISBEHAVING
CHANGE IN FILE SIZE
YES
SENSITIVE DATA
NO
SERVER CHECKS FOR AUTHORIZATION
SERVER SENDS NOTIFICATION TO OWNER
USER RETURNS DOCUMENT
YES
OWNER CHECKS FOR AUTHORIZATION
NO
GENERATES RE-ENCRYPTION KEY AND SEND TO SERVER
SERVER SENDS RE-ENCRYPTION KEY TO USER
STOP

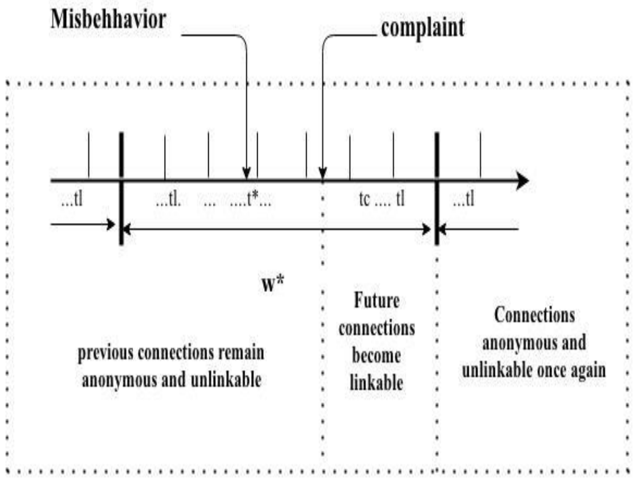
Fig 2 Data Flow Diagram

B. Blacklisting a user

If a user misbehaves, the server may link any future connection from this user within the current linkability window (e.g., the same day). A user connects and misbehaves at a server during time period t∗ within linkability window w∗. The server later detects this misbehavior in time period tc (t∗ < tc ≤ tL) of the same linkability window w∗. As part of the complaint, the server presents the nymble ticket of the misbehaving user and obtains the corresponding seed tc, tc from the NM. The server is then able to link future connections by the user in time periods + 1,...,tL of the same linkability window w∗ to the complaint. Therefore, once the server has complained about auser, that user is blacklisted for the rest of the day, for example (the linkability window). Note that the user's connections in t1, t2,...,t∗, t∗ + 1,...,tc remain unlinkable (i.e., including those since the misbehavior and until the time of complaint). Even though misbehaving users can be blocked from making connections in the future, the users' past connections remain unlinkable, thus providing backward unlinkability and subjective blacklisting

c. Notifying user of blacklist status

Users be notified of their blacklist status before they present a nymble ticket to a server. In our system, the user can download the server's blacklist and verify her status. If blacklisted, the user disconnects immediately. If a user is blacklisted, nymble server sends notification to user. This notification is send by using short message service. Since the blacklist is cryptographically signed by the nymble server the authenticity of the blacklist is easily verified if the blacklist was updated in the current time period (only one update to the blacklist per time period is allowed). If the blacklist has not been updated in the current time period, the NM provides servers with "daisies" every time period so that users can verify the freshness of the blacklist ("blacklist from time period told is fresh as of time period tnow"). The daisies are elements of a hash chain, and provide a lightweight alternative to digital signatures. Using digital signatures and daisies, we thus ensure that race conditions are not possible in verifying the freshness of a blacklist. A user is guaranteed that he or she will not be linked if the user verifies the integrity and freshness of the blacklist before sending his or her nymble ticket.


Misbehhavior
complaint
...tl
...tl. ... .t*...
tc .... tl
...tl
w*
previous connections remain anonymous and unlinkable
Future connections become linkable
Connections anonymous and unlinkable once again

D. ASCII Encryption

Cryptography is the branch of art and science of keeping secret messages secret by which unauthorized users can't access that secret message. So, to make message secret we use encryption i.e., convert plain-text or message into some other code called cipher -text and decryption i.e. convert that cipher-text into original readable form. For encryption and decryption we need two party sender and receiver. The given input on which encryption is to be done and obtain ASCII value for it. Then perform ASCII code to numerical value conversion. Their respective numeric code will be converted into ASCII characters. This conversion is very simple. Then add the character into the command window and press the ok button. After that automatically gives out the resultant numeric code. From the generated substitution array Substitute the numbers sequentially on a one-to-one basis at the place of character of the plain text. Now divide the number chosen with the ASCII value of the plain text character and calculate the Quotient and the Remainder and store it in the Array. The message will now be the series of Quotient followed by the Remainder. A substitution array approach used ASCII values for the encryption and decryption process. ASCII stands for American standard code for information inter change has been adopted by several American computer manufactures as their computer's internal code American standard association developed ASCII. ASCII value of each character is different. As A-65, B-66, C-67 or a -97, b-98 and so on.

Basic Terminology

Plain text : The original intelligible message.

Cipher text : The transformed or coded message.

Cipher : An algorithm for transforming an intelligible message into one**.**

Key: Some critical information used by cipher.

For Encryption: Y = EK(X)

Where Y=cipher text, E= encryption, K= key, & X= plain text

For Decryption: X= DK(Y)

Where X= plain text, D= decryption, K= key, Y= cipher text

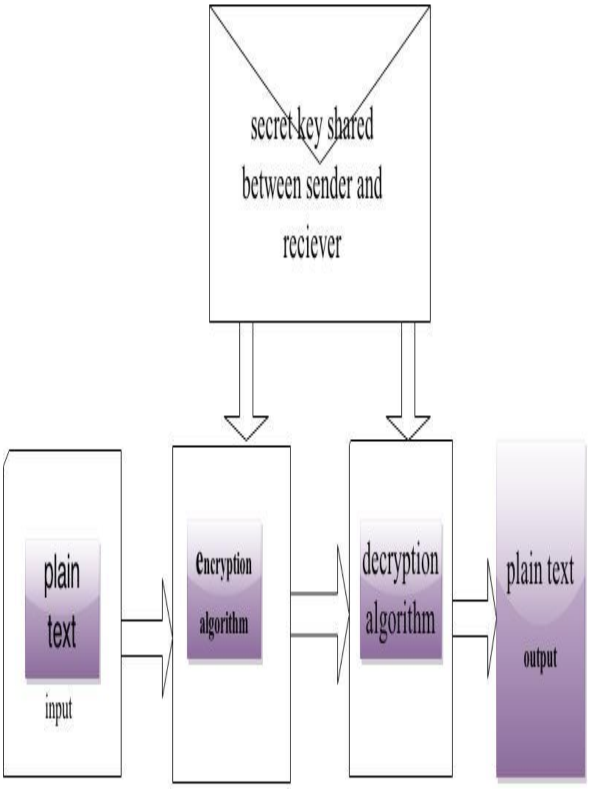
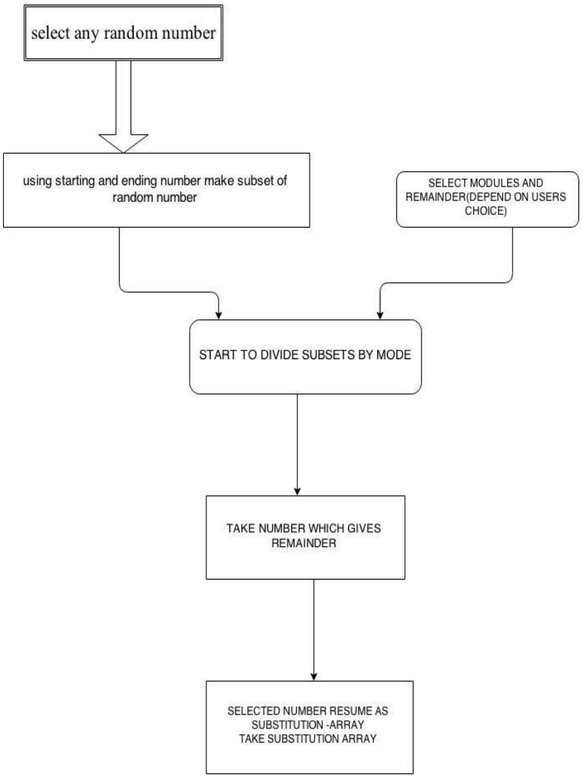


Fig:Conventional encryption model

E. ARCHITECTURE

The architecture consists of the following steps:Firstly select any number randomly.After selection use starting and ending number and make subset, followed selection of modulus and remainder as well.When subset is selected then it is divided by mode. After division take only those number which gives remainder. Finally selected numbers will be resumed as substitution array.


select any random number
using starting and ending number make subset of random number
SELECT MODULES AND REMAINDER(DEPEND ON USERS CHOICE)
START TO DIVIDE SUBSETS BY MODE
TAKE NUMBER WHICH GIVES REMAINDER
SELECTED NUMBER RESUME AS SUBSTITUTION -ARRAY
TAKE SUBSTITUTION ARRAY

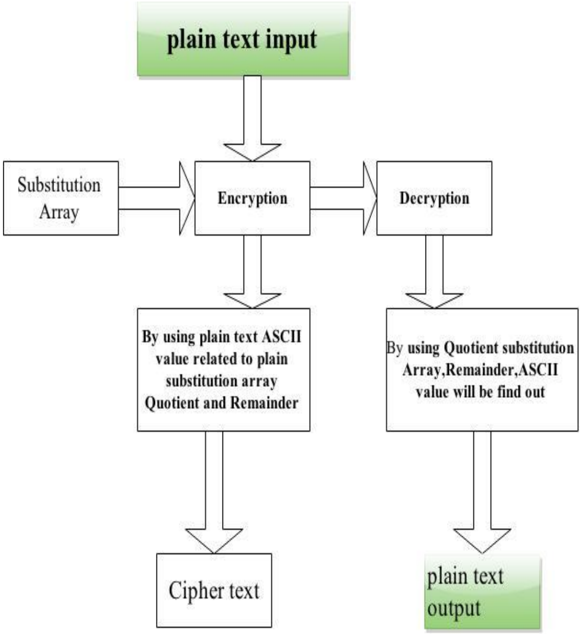

plain text input
Substitution Array
Encryption
Decryption
By using plain text ASCII value related to plain substitution array Quotient and Remainder
By using Quotient substitution Array,Remainder,ASCII value will be find out
Cipher text
plain text output

Fig : ENCRYPTION AND DECRYPTION

V.CONCLUSION

Cloud computing is a distributed system in, where different users in different domains can share data among each other. Different Identity-based proxy encryption and re-encryption schemes have been proposed to outsource sensitive data from the owner to an external party. Nevertheless, they cannot be employed in cloud computing. As security of data storage is important, also the security of data transfer is important. We enhance the security of data transfer by introducing the identity based secure encryption and re-encryption. In our paper files are classified into sensitive file and nonsensitive file. Files are encrypted uder owners identity and ASCII encryption is used. After encryption owner outsource his files to Nymble server.Before outsourcing owner who wishes to use service by Nymble server have to register with Nymble server. For non sensitve file, authorization is done by server itself but for sensitive data owner decides the permission. User have to authorize himself to nymble server and owner using his id and password.It will provide many advantages like collusion-resistance over the previous schemes, block misbehaving user and will get the notification of user request on his/her mobile phone and will also provide security against Distributed Denial of Service attack and it provides secure model of cloud storage with safe data forwarding. User can request for service by nymble server only after registering. After using file the user have to send back the file to nymble server. Their nymble checks the copy returned by user with original copy given to it bu owner. If there is change in file size then it means user has misbehaved. So the nymble server blacklist the user and notifies him about it. For next 48 hours he cannot access the nymble server.

REFERENCE

[1] Jinguang Han,Student Member, IEEE, Willy Susilo,Senior Member, IEEE, andYi Mu,Senior Member, IEEE," Identity-Based Secure Distributed Data Storage Schemes" IEEE TRANSACTIONS ON COMPUTERS, VOL. 63, NO. 4, APRIL 2014

[2]. Varsha S.Agme ,Prof. Archana C.Lomte," Cloud Data Storage Security Enhancement Using Identity Based Encryption" International Journal of Application or Innovation in Engineering & Management (IJAIEM)

[3].Patrick P.Tsang,Apu Kapadia,Cory Cornelius and Sean W.smithk "Nymble: Blocking misbehaving users in anonymizing networks" Dartmouth Computer Science Technical report TR2008-637

[4].Mathew Green,Giuseppe Ateniese "Identity Based Proxy Encryption" Information Security Institute,Johns Hopkins University.

[5].Abdul Wahid Khan,Siffat Ullah Khan,Muhammad Ilyas and Muhammad Ilyas Azeem"A Literature survey on Data Privacy/Protection Issues and Challenges in Cloud computing "IOSR Journal of computer Engineering (IOSRJCE)

[6].P.Sahithi Chaitanya,M.murali "Improved Schemes to Secure Distributed Data Storage against Untrusted Users" IJCSIT

[7].Sangeeta Solanki, Bijnor, Muzaffar Shobhit ,Vineet Sukhraliya "Encryption and Decryption Algorithm using ASCII values with substitution array Approach" International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 8, August 2013.