Journal of Theoretical and Applied Information Technology
30th June 2016. Vol.88. No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SECURING DATA STORED IN CLOUDS USING MULTI KEYS AND PROXY INJECTION SCHEMES

[1]GIRISHMA.V, [2]Dr. K.V.V. SATYANARAYANA

[1]M.Tech, K L University, Vijayawada
[2]PROFESSOR, K L University
E-mail: [1]gvenna6@gmail.com, [2]kopparti@kluniversity.in

ABSTRACT

A new robust control scheme for Multi key distribution scheme that supports secured data storage and access in clouds along with anonymous upload feature to protect user privacy is proposed.User authenticity is established by the cloud through proper registration procedures and in turn data authenticity with multi key sharing authenticity and support for anonymous sharing is established by registered users. Access control is being implemented where the stored information can be decrypted by users who are valid.Replay attacks are prevented through Proxy injection based schemes and they are also helpful in containing Cloud Services Provider (CSP) from knowing where-about of uploader themselves. User revocation is addressed and creating, reading and modifying information in cloud is also supported.
Keywords: *Cloud Data, Attribute Based Encryption, Anonymity Based Uploads and Key Distributions.*

1. INTRODUCION

In cloud computing concept, computation and also storage are outsourced to servers by utilizing the Internet. This supports users by avoiding issues relating to maintenance of resources online. In clouds, highly sensitive data is stored mostly, example, health records and social networks information. Hence preservation and confidentiality are most important in cloud computing. On one side, authentication of user is required before proceeding with any transaction, and on the other side, tampering of outsourced data by cloud should not happen.

Identity of the user should be kept private thus ensuring user privacy to other clouds or users. The cloud should be held accountable to services it is providing. . User validity is verified as to keep track of who is storing data into cloud .law enforcement is also implemented along with technicalities to make sure of privacy and security. Data modification issues and server colluding attacks may happen to cloud. In the latter attack, storage servers are compromised by adversary, which leads to data files modification which is consistent internally. Hence for secure storage of data, encryption has to be done.

Performing an efficient inquiry on encrypted data is also prominent criteria in cloud environment. The query should be hidden by clouds but satisfying records should be returned. Through searchable encryption, this can be achieved.

The keywords are pushed in encrypted format to the cloud, and in response cloud sends back the result without the knowledge of original keyword on which search is performed. But here data records should be associated with keywords in order for a successful search. The identical records are retrieved only when keywords are similar. Homomorphic cipher methods are embedded to ensure that cloud is not tracking or reading our data when performing computations on it. Using these cipher text of the data is received by cloud and calculations are performed on the encoded text and encrypted value of the response is sent back. The user can unencrypt the result, and has to verify whether cloud returned correct result. By this way cloud can't know the data on which it has been operated. Accountability of clouds is important any operations performed by cloud or requested by users should not be denied. Log should be maintained and information in log should be controlled[10].

Entry control in cloud environment is becoming prominent because only legitimate users should have permission to prescribed services.

The types of entry control mechanisms are available: user based access control system UBAC, role-based access control system RBAC and attribute-based Access control system ABAC. In

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

UBAC, the users who are legitimate to use data are contained in the access control list as there will be several users, this mechanism is not appropriate for cloud environment.

In Role Based Access Control [1,2,3,4,5,6] based on their individual roles, users are classified. Users who have matching roles can access the data. System defines the roles. For example a virtual member hierarchy is implemented here where roles such as R = {Admin, Manager, Staff, Support Staff, Experience, Seniority} are defined as possible roles and set of actions A = {View Data, Edit/Update Data, Manage Users, Assign roles, Define-Data-Access-Policies with access modes (Private,Public,Specific)} are prescribed such that for. In such a scenario a user with administrative access to the data defines an auto generated access policy on users matching a set of attributes corresponding to their roles and actions. For instance, an access policy can be generated for a user who is a Staff with more than 5 years experience and who has editing privileges over the data. Areas where usage control is prominently utilized are health care and social networking. In healthcare, to save personal details about patients clouds are used so that data can be accessed only by legitimate users. In social groups online to share private details, photos, videos of people with selected groups, access control is used anonymity of the user also should be considered along with storing the contents of data securely. The user without revealing his identity should confirm that he is the actual owner who uploaded the information. Cryptographic protocols named ring, mesh and group signatures are available to utilize. Ring signature is not an appropriate option as there would be several users in cloud environments. Group signature option has an already existing group and hence not possible in cloud environment. Mesh signature technique do not ensure if the message is from single or from multiple users' collaborating together. Hence a new mechanism named attributes based signature technique is introduced. Here an authenticity tag (Numeric Hash) is appended to the generated key when the owner of the file grants access to a receiver via a key sharing process. So during access and decryption phase it helps to establish that the accessing user is actually authorized or not without disclosing the owners identity on the process. So this appended authenticity tag (AAT) in combination with an Attribute Based Encryption schemes results in an identity disclosing proofed data access control. In a traditional data securing approach a single key phrase specified by the owner of the file at the time of uploads needs to be distributed with relevant receivers intended by the owner. Here the uniqueness of the key phrase can be compromised when reveled with more than one user and can lead to guessing/leakage attacks with respect to other files or other users. So the idea of a multi key generation scheme with a bounding threshold for each user and for each file is developed here to sustain that key uniqueness factor to reduce the complexity involved in validation and authenticity establishing schemes.

In decentralized approach, users who want to be anonymous are not authenticated. Also in Mechanism of distributed access control, user authentication is not provided and also modify access is not allowed to anyone other than the owner. Hence earlier version of this paper is enhanced to add up verification of authenticity of message .Replay attacks are also handled so users can alter stale data by replacing it with fresh data even though user is revoked of attributes.

The main points contributed in this paper are as follows:

1. Only legitimate users with appropriate attributes can use data in cloud through Distributed access control approach
2. User authentication and anonymity of user who stores and modifies their data on the cloud.
3. Decentralized architecture: maintaining multiple KDCs for key management.
4. The usage control and validation are collaboratively resistant in nature, means that no two users can collaborate and use data or authenticate on their own, without individual authorization
5. Once revoked, those users cannot access data
6. Stale information can't be written back by user whose properties and keys are been revoked. Hence making this scheme resilient to replay attacks.
7. Several read and write operations on the stored cloud data are supported by protocol
8. Extensive costs related to pairing and other calculation costs are equivalent to centralized techniques.

Journal of Theoretical and Applied Information Technology
30th June 2016. Vol.88. No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2. LITERATURE REVIEW

Originally the theory of Encryption based on attributes also called ABE was used in order to secure information with respect to cloud computing domain. In this access policy users tend to define the attributes that are required to construct a public key that will be shared among users which can be used to encrypt or decrypt contents. The attributes range from Data Owners identity (involving credentials) and file Meta data that includes name, extension, size, creation date etc. prior approaches implemented a centralized approach which happens to be a monotonic structure to generate and handle key distributions[8].

Multi authority based solutions involving ABE also proposed earlier which is void off a trusted authority[9.]

In all these approaches decryption at receiver's end is computationally an expensive process. An original design that involves a decentralized approach was proposed by Maji helps us to understand and implement multiple KDC's better. But that solution is susceptible to replay attacks[11].

3. BACKGROUND WORK

Our proposed implementation requires the following assumptions:

1) The cloud service provider (CSP) is assumed to be honest but at the same time happens to be curious. In other words they can view user's content even though they can't modify it. This honest but curious CSP's can't in any way tamper the data and hence it is not a threat to user's cloud storage operations.

2) Users can have either read (download) or write (delete) access to their contents in cloud.

3) Although we demonstrated using a local HTTP protocol, the usual communication between user's and CSP are secured by Secure Shell protocol (SSH) protocol

Structure of User Access Policies

Three Types were proposed with respect to the structure of access policies

1) Boolean outcomes for attributes (Not used in our system)

2) Linear secret sharing scheme (LSSS) matrix (used in our current implementation)

3) Monotone access structures combined with Boolean functions (not used in our current system)

4. PROPOSED WORK

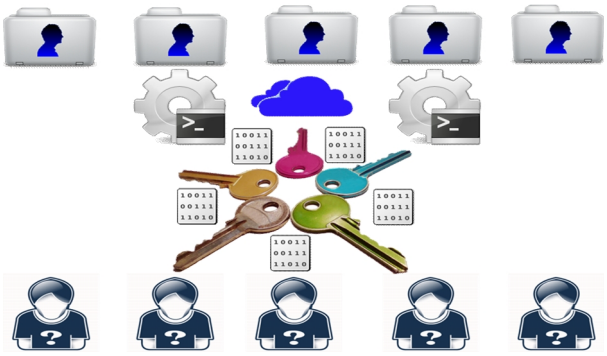

Figure 1: Architecture Flow With Anonymity Factor And M-KDC Concepts

The encryption function we define is Encrypt_ File (owner, receiver, AAT, file). Data Owner selects the access tree 'X' consisting of attributes. An LSSS matrix 'R' can be concluded using the following steps.

Data Owner encryption procedure is as follows:

1) Choose a random seed and a vector indicating a Data Owner
2) Estimate attribute size
3) Obtain Data Owner credentials which happens to be primary attribute
4) Obtain file to be shared (Upload)
5) Data Owner selected upload anonymity (optional)
6) Data Owner specified key threshold
7) Data Owner specified total keys
8) Data Owner specified proxy key
9) Generate secondary attributes from file Meta data.
10) Generate perceived mappings of key value pair from a constant ABE key generated from the above process and a random seed of size N defined by the user

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

11) Encrypt the file using the above generated ABE key
12) Share it with an intended receiver

Proxy Injections to Anonymity from Server

To achieve anonymity even from the server we also propose to incorporate the following steps

Proxy Initiations

The first stage is to create a proxy on local server. Later we initiate HTTP calls to the local server and retrieve dynamic proxy data from the remote server. This one sustains without failure because the application module runs on the local web medium server and is not subject to Same Origin Policy also called SOP restrictions. Only the client-side modules are subjected to those validations. Alternately we can load JavaScript through a <script> tag using online fashion. Client side JavaScript has the capability to maintain and manipulate <script> tags in the HTML Document Object Model also called DOM.

Client mode can set the src attribute of a <script> tag to automatically invoke new js script into the page. This technique is not prone to SOP rules and hence we can effectively use it to load JavaScript dynamically. An algorithmic implementation to create and implement a new proxy policy is as follows which involves deliberate manipulation of up loader credentials with dynamic data injection schemes rendered on the fly that can suppress original owners credentials and replace it with a proxy identity.

Algorithm 1 PolicyCompare

Input: new policy (M', ρ') with $l' \times k'$ matrix
Input: previous policy (M, ρ) with $l \times k$ matrix
Output: $I_{1,M'}, I_{2,M'}, I_{3,M'}$ ▷ three subsets of row indexes in M'

```
I_M ← index set of rows in M
for j = 1 to l' do
    if ρ'(j) in M then
        if I_M != ∅ & ∃i ∈ I_M s.t. ρ(i) == ρ'(j) then
            add (j,i) into I_{1,M'}
            delete i from I_M
        else
            find any i ∈ [1,l] s.t. ρ(i) == ρ'(j)
            add (j,i) into I_{2,M'}
        end if
    else
        add (j,0) into I_{3,M'}
    end if
end for
```

It initially makes a call to the policy correlating algorithm Policy Compare to correlate the new access technique with the earlier one, and returns 3 sets of row indexes which are shuffled to create a perturbed access policy which cannot be reconstructed by the server but yet stored at the server.

It adapts based on the owner, receiver, content attributes along with Multiple key distributions generated for the data content. Considering its dynamic efficient nature while upholding privacy and security with respect to cloud data storages it is a much better system compared to prior approaches.

Decryption Steps Implemented At Receiver Side

The decryption function is Decrypt File (file, keysSet). Receiver X takes an input of incoming file and generates an ABE key using file metadata, self and Data Owners credentials which is denoted as ABE key and the encrypted file as cipher text C

Steps involving the above procedure are as follows

1) X receives an incoming file
2) A client module at X checks for user's permission
3) X derives ABE key from receiver S credentials and file F metadata
4) X generates a threshold based public key from ABE key.
5) X decrypts received file from the above key mapping pair.

5. PERFORMANCE ANALYSIS

W-M-R: single person can write and several people can read. M-W-M-R- several users can both read and write. Our distributed scheme supports many writes and is robust and decentralized, supports privacy preserving authentication, support user revocation, computation load and connection costs spent by the users and cloud environment are comparable to centralized approaches

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

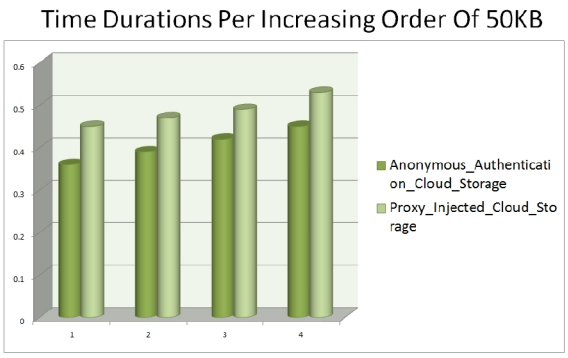

Figure 2: Time Durations Per Increasing Order Of 50KB

A comparative chart with respect to cloud uploads in with respect to anonymity and proxy injection with increasing orders of size 50kb is plotted here. With a slight increase in time complexity we could attain anonymity even from CSP themselves.

6. CONCLUSION

Here we have demonstrated an access control technique using decentralized key distribution providing anonymous authentication, user revocation preventing replay attacks. The cloud environment validates the user credentials without having the knowledge of user identity storing information in cloud. But even in this, cloud environment will have the idea about the access strategy for each record saved in cloud which we strive to overcome using proxy injection techniques and its efficiency is highlighted.

REFERENCES

[1] "Role-Bases Access Controls", 1992, D.F.Ferraiolo and D.R.Kuhn.
[2] "Adding Attributes to Role Based Access Control", 2010, D.R.Kuhn, E.J.Coyne and T.R.Weil.
[3] "Securing Personal Health Records in Cloud Computing: Patient-Centric and Fine-Grained Data Access Control in Multi-Owner Settings", 2010, M.Li, S.Yu, C.Wang, K.Ren and W.Lou.
[4] "Attribute Based Data Sharing with Attribute Revocation", 2010, S.Yu, C.Wang, K.Ren and W.Lou
[5] "Hierarchical Attribute-Based Encryption for Fine-Grained Access Control in Cloud Storage Services", 2010, G.Wang, Q.Liu and J.Wu.
[6] "Realizing Fine-Grained and Flexible Access Control to Outsourced Data with Attribute-Based Cryptosystems", 2011, F.Zhao, T.Nishide and K.Sakurai.
[7] "DACC: Distributed Access Control in Clouds", 2011, S.Ruj, A.Nayak and I.Stojmenovic.
[8] "Multi-Authority Attribute Based Encryption", 2007, M.Chase
[9] 9."Decentralizing Attribute-Based Encryption", 2011, A.B.Lewko and B.Waters