An Identity Based Authentication and Data Encryption in Cloud Computing

Sravan Kumar Nalla[1], Konni Srinivasarao[2]
Final M.Tech Student[1], Asst.professor[2]
[1,2]Dept of CSE, *Sarada Institute of Science, Technology and Management (SISTAM), Srikakulam, Andhra Pradesh*

Abstract:

Now a days security place an important role for data privacy in a cloud computing. Moving on to data security in the cloud usually implies relying on the cloud service provider for protection. The cloud service provider could potentially access the data or even provide it to third parties. More ever one should trust the cloud service provider to legitimately apply the access control rules defined by the data owner for other users. In the cloud computing one of the problems is transfer data from one cloud service provider to another cloud service provider. Because to share data from more than one cloud service provide is face the problem of data security. In the cloud computing is another problem is authorization of users in the cloud. So that by overcome those problems we can implement a novel multi authority identity based technique for authentication of users in the cloud and an efficient cryptography technique for data encryption and re encryption of data in the cloud. Another enhancement of this paper is proposed authorization model with more action like modification and deletion. By implementing those techniques we can provide more efficiency of in a novel network and also provide more security of transferred data in the cloud.

Keywords

Authentication, Key Generation, Mater Key, Cryptography, Message Digest, Signature, Random Nonce.

I. INTRODUCTION

Security is one of the primary client attentiveness toward the reception of Cloud registering. Moving information to the Cloud typically infers depending on the Cloud Service Provider (CSP) for information insurance. In spite of the fact that this is normally overseen based on lawful or Service Level Agreements (SLA), the CSP could possibly get to the information or even give it to outsiders. In addition, one ought to believe the CSP to honestly apply the get to control rules characterized by the information proprietor for other clients. The issue turns out to be much more unpredictable in Inter-cloud situations where information may spill out of one CSP to another. Clients may misfortune control on their information. Indeed, even the trust on the unified CSPs is outside the control of the information proprietor. This circumstance prompts to revaluate about information security approaches and to move to an information driven approach where information are self-ensured at whatever point they live. Encryption is the most generally utilized technique to ensure information in the Cloud. Truth be told, the Cloud Security Alliance security direction prescribes information to be ensured very still, in movement and being used [1]. Encoding information maintains a strategic distance from undesired gets to. Notwithstanding, it involves new issues identified with get to control administration. A run based approach would be attractive to give expressiveness. In any case, this assumes a major challenge for an information driven approach since information has no calculation abilities independent from anyone else. It is not ready to authorize on the other hand figure any get to control lead or strategy. This raises the issue of arrangement choice for a self-secured information bundle: who ought to assess the guidelines upon a get to ask? The to start with decision is have them assessed by the CSP, yet, it could conceivably sidestep the standards. Another choice is have rules assessed by the information proprietor, however this infers that either information couldn't be shared or the proprietor ought to be online to take a choice for every get to ask.

Security is one of the main user concerns for the adoption of Cloud computing. Moving data to the Cloud usually implies relying on the Cloud Service Provider (CSP) for data protection. Although this is usually managed based on legal or Service Level Agreements (SLA), the CSP could potentially access the data or even provide it to third parties. Moreover, one should trust the CSP to legitimately apply the access control rules defined by the data owner for other users. The problem becomes even more complex in Inter cloud scenarios where data may flow from one CSP to another. Users may loss control on their data. Even the trust on the federated CSPs is outside the control of the data owner. This situation leads to rethink about data security approaches and to move to a data-centric approach where data are self-protected whenever they reside. Encryption is the most widely used method to protect data in the Cloud. In fact, the Cloud Security Alliance security guidance recommends data to be protected at rest, in motion and in use. Encrypting data avoids undesired accesses. However, it entails new issues related to access control management. A rule-based approach would be desirable to provide expressiveness. But this supposes a big challenge for a data-centric approach since data has no computation capabilities by itself. It is not able to enforce or compute any access control rule or policy. This raises the issue of policy decision for a self-protected data package: who should evaluate the rules upon an access request? The first choice would be to have them evaluated by the CSP, but it could potentially bypass the rules. Another option would be to have rules evaluated by the data owner, but this implies that either data could not be shared or the owner should be online to take a decision for each access request.

II. RELATED WORK

Different approaches can be found in the literature to retain control over authorization in Cloud computing. In [2] authors propose to keep the authorization decisions taken by the data owner. The access model is not published to the Cloud but kept secure on the data owner premises. However, in this approach the CSP becomes a mere storage system and the data owner should be online to process access requests from users. Another approach from [3] deals with this issue by enabling a plug-in mechanism in the CSP that allows data owners to deploy their own security modules. This permits to control the authorization mechanisms used within a CSP. However, it does not establish how the authorization model should be protected, so the CSP could potentially infer information and access the data. Moreover, this approach does not cover Inter-cloud scenarios, since the plug-in module should be deployed to different CSPs. Additionally, these approaches do not protect data with encryption methods. In the proposed SecRBAC solution, data encryption is used to prevent the CSP to access the data or to release it bypassing the authorization mechanism. However, applying data encryption implies additional challenges related to authorization expressiveness. Following a straightforward approach, one can include data in a package encrypted for the intended users. This is usually done when sending a file or document to a specific receiver and ensures that only the receiver with the appropriate key is able to decrypt it. From an authorization point of view , this can be seen as a simple rule where only the user with privilege to access the data will be able to decrypt it (i.e. the one owning the key). However, no access control expressiveness is provided by this approach. Only that simple rule can be enforced and just one single rule can apply to each data package. Thus, multiple encrypted copies should be created in order to deliver the same data to different receivers. To cope with these issues, SecRBAC follows a data-centric approach that is able to cryptographically protect the data while providing access control capabilities. Several data-centric approaches, mostly based on Attribute-based Encryption (ABE) [4], have arisen for data protection in the Cloud [5]. In ABE, the encrypted cipher text is labelled with a set of attributes by the data owner. Users also have a set of attributes defined in their private keys. They would be able to access data (i.e. decrypt it) or not depending on the match between cipher text and key attributes. The set of attributes needed by a user to decrypt the data is defined by an access structure, which is specified as a tree with AND OR nodes. There are two main approaches for ABE depending on where the access structure resides: Key-Policy ABE (KP-ABE) [4] and Cipher text-Policy ABE(CP-ABE) [6]. In KPABE the access structure or policy is defined within the private keys of users. This allows encrypting data labelled with attributes and then controlling the access to such data by delivering the appropriate keys to users. However, in this case the policy is really defined by the key issuer instead of the encrypt or of data, i.e. the data owner. So, the data owner should trust the key issuer for this to properly generate an adequate access policy. To solve this issue, CPABE proposes to include the access structure within the cipher text, which is under control of the data owner. Then, the key issuer just asserts the attributes of users by including them in private keys. However, either inKPABE or CP-ABE, the expressiveness of the access control policy is limited to combinations of AND-ed or attributes. The data-centric solution presented in this paper goes a step forward in terms of expressiveness, providing a rule-based approach following the RBAC scheme that is not tied to the limitations of current ABE approaches.

III.PROPOSED SYSTEM

In the proposed system we are implementing an efficient novel multi authority identity based technique for authentication of data consumers. Before performing authentication of users or data consumers each user will register into cloud. After completion of user registration the cloud server will performing authentication process. The completion of authentication process the cloud server will generate master key for all users. The master key is used for encryption and re encryption of shared data in cloud. After successful authentication of each user the cloud server will send the master key to all users in the cloud. The cloud server will also send the master key to data owner. The cloud server will send mater key to respected users mail ids. After completion of key generation process the data owner will encrypt the data and stored into cloud. The data consumer or user will retrieve required file and re encrypt that file. After completion of re encryption process the user will get original plain format data. The implementation procedure of multi authority identity based technique is as follows.

A) Authentication of Users:

In this module each user will be identify by the cloud server by performing multi authority identity based technique. Before performing authentication process each user will login into cloud server. After completion of log in process each user will perform the following steps for performing authentication process.

1. Each user or data consumer randomly choose the random nonce R_i and send that value to cloud server.

2. The cloud server will retrieve all users random nonce and generate universal key to all users. After generating that key the cloud server will send universal key (U) to all users in the cloud.

3. Each user will retrieve universal key and generate shared point (x_i, yi). the user will take shared point and perform xor operation between universal key and shared point is (x_i, y_i® U). Take the xor shared point and send to cloud server.

4.The cloud server will retrieve all users xor shared points and get original share points (x_i,y_i) by performing xor operation. The cloud server takes those shared points of all users and generates unique identity of each user by using following equation.

$(u_{id})_I$ = Ri ® U ® (xi, yi)

5. After generating unique identity of each user the cloud server will send those ids to individual users.

6. Each user will retrieve the unique identity and generate signature by using following equation.

sig_i = H (id ® R_i® U ® (x_i, yi) ® $(u_{id})_i$)

7. After generating signature each user will send those signatures to cloud server. The cloud server will retrieve all users signature and again generate signature by using those values.

8. The cloud server will take both signature and compare it. If the both signatures are equal those users are authenticated users else those users are not authenticated users.

After completion of authentication process the cloud server will sending authentication status of individual users. Before sending authentication status of users the cloud server will generate master key for encryption and decryption process. After generating master key the cloud server will send that key to all users mail id. The implementation procedure of master key is as follows.

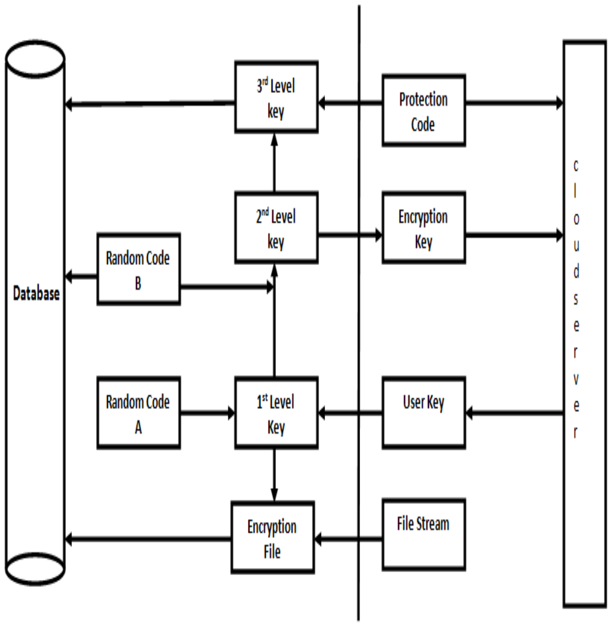
B) Generation of Master key:

In this module the cloud server will generate master key and send that key all users in the cloud. The sending of master key can be done only the authenticated users in cloud. The master key can be send through respect user's mail ids. The cloud server will also send master key to data owner. The data owner will use master key for encryption of uploaded documents. The generation of master key is as follows.

1. In the key generation process the cloud server will randomly generate code A and the string in request stream is encoded with code A to generate the first level encryption key. The first level encryption key is also known as master key.
2. In the second level, the cloud server randomly generates code B and the first level encryption key is encoded with code B to generate the second level encryption key.

3. The code B is stored in the database, and the new file of second level encryption key is generated and sent to the each user by using the mail which is developed by using the smtp protocol.

4. In case of losing the second level encryption key, the cloud server generates the third level encryption key based on the second level encryption key and a protection code which is randomly generated by the system. The third level encryption key is stored in the database.



The completion of key generation process the data owner will retrieve first level encryption key and encrypt the uploaded file. After completion of encryption process the data owner will store the cipher format document into cloud server. The cloud server will contain all cipher formatted documents. So that nobody cannot get plain format documents. So that we can provide privacy of uploaded documents in the cloud. The implementation process of data encryption is as follows.

C) Encryption Process:

In this module the data owner will choose upload document and perform the encryption process. In this paper we are using extended tiny encryption algorithm. The implementation procedure of extended tiny encryption algorithm of encryption process is as follows.

```
          private final static int SUGAR = 0x9E3779B9;
          private final static int CUPS = 32;
 void brew (int[] buf)
{
 assert buf.length % 2 == 1;
 int i, v0, v1, sum, n;
 i = 1;
while (i<buf.length)
  {
   n = CUPS;
  v0 = buf[i];
  v1 = buf [i+1];
  sum = 0;
 while (n-->0)
  {
v0 += (((v1 << 4 ) ^ (v1>>5)) + v1) ^ (sum +S[sum & 3]);
sum += SUGAR;
v1 += (((v0 << 4) ^ (v0 >> 5))+ v0) ^ (sum +S[(sum>>11) & 3]);
  }
buf[i] = v0;
buf[i+1] = v1;
i+=2;
}
}
```

The completion of encryption process the data owner will stored cipher format into cloud server. After completion storing process each user will get all cipher format document and select required document. Take that document and perform the decryption process of extended tiny encryption algorithm. Before performing decryption process each user will be authenticated by cloud server. After successful authentication of users each user will get second level and get the first level encryption key or master key. Before getting master key each user will enter the second level key and get first level key or master key. Take that key and perform the decryption process.

D) Decryption Process:

In this module each user will retrieve the first level encryption key and perform the decryption process. In this module the user will take second level and using that key user will get first level encryption key. After getting first level encryption key the user will choose required document and perform the decryption process. The implementation procedure of decryption process is as follows.

```
    private final static int SUGAR = 0x9E3779B9;
    private final static int CUPS = 32;
    private final static int UNSUGAR = 0xC6EF3720;
       void unbrew (int [] buf)
       {
           assert buf.length % 2 == 1;
           int i, v0, v1, sum, n;
           i = 1;
           while (i<buf.length)
{
  n = CUPS;
  v0 = buf[i];
  v1 = buf[i+1];
  sum = UNSUGAR;
  While (n--> 0)
{
v1 -= (((v0 << 4) ^ (v0 >> 5))+ v0) ^ (sum +S[(sum>>11) & 3]);
sum -= SUGAR;
v0 -= (((v1 << 4) ^ (v1>>5)) + v1) ^ (sum +S[sum & 3]);
  }
buf[i] = v0;
buf [i+1] = v1;
i+=2;
 }
}
```

After completion of decryption process each user will get original plain format documents. In this paper we are also implement the modification and deletion of users in a cloud. By performing those operations we can improve efficiency of authentication process and also provide more security of sharing data in the cloud.

IV. CONCLUSIONS

In this paper we are proposed a novel multi authority and efficient cryptography techniques for authentication of users or data consumers, privacy of shared data in the cloud. Before stored data into cloud server each user will be identify by the cloud server. After completion of authentication process the cloud server will generate master key or first level encryption key for data encryption and decryption. Before performing encryption and decryption process the cloud server will send second level key all user mail ids. By using second level each user will get first level encryption key. The cloud server also sends first level encryption key or master key to data owner. The data owner will take first level encryption key and perform the encryption process. In this paper we are using extended tiny encryption algorithm for performing encryption process. After completion of encryption process the cloud server will stored cipher formatted data into cloud server. The authenticated user will take second level key and get the first level encryption key. By using first level encryption key or master will perform the decryption process of extended tiny encryption algorithm. After completion of decryption process each user will get plain format data. By performing those operations we can improve privacy in the shared data and provide more efficiency in the authentication, key generation process.

REFERENCES

[1] Cloud Security Alliance, “Security guidance for critical areas of focus in cloud computing v3.0,” CSA, Tech. Rep., 2003.

[2] A.Lawall, D. Reichelt, and T. Schaller, “Resource management andauthorization for cloud services,” in Proceedings of the 7th International Conference on Subject-Oriented Business Process Management,ser. S-BPM ONE ’15, New York, NY, USA, 2015, pp. 18:1–18:8.

[3] D. Y. Chang, M. Benantar, J. Y.-c. Chang, and V. Venkataramappa, “Authentication and authorization methods for cloud computing platform security,” Jan. 1 2015, uS Patent 20,150,007,274.

[4] B. B and V. P, “Extensive survey on usage of attribute based encryption in cloud,” Journal of Emerging Technologies in Web Intelligence,vol. 6, no. 3, 2014.

[5] B. Waters, “Cipher text-policy attribute-based encryption: An expressive, efficient, and provably secure realization,” in Public Key Cryptography - PKC 2011, 2011, vol. 6571, pp. 53–70.

[6] O. K. J. Mohammad, S. Abbas, E. M. El-Horbaty, and A. M. Salem, “Innovative method for enhancing key generation and management in the aes-algorithm,” CoRR, vol. abs/1504.03406,
2015.

[7] Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” in Proceedings of the 13th ACM Conference on Computer and CommunicationsSecurity, ser. CCS ’06, New York, NY, USA, 2006, pp.89–98.