International Journal on Applications in Information and Communication Engineering
Volume 1: Issue 7: July 2015, pp 21-24. www.aetsjournal.com ISSN (Online) : 2394-6237

Efficient Data Sharing in Cloud Medium with Key Aggregate Cryptosystem

A. Shakila Banu,

Abstract— In cloud storage the Resources are shared and, remotely accessed. And then we show how to securely, professionally, share data with others in cloud storage. And consider Aggregation is a collection of things to make retrieve a data but the key aggregate is a multiple keys produce a set of result in the cryptosystem. In the Advanced Key sharing system based on hint text methodology is formed to share the data safely.Once the data sharing is completed then the key aggregate differs from its actual form. So the user cannot guess the key aggregate cryptosystem and this process provides efficient solution than the existing ones.

Keywords— Cloud storage, Information Retrieval, Hint Text Manipulation, Dynamic Decryption.

I. INTRODUCTION

Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network. Cloud providers manage the infrastructure and platforms on which the applications run. Cloud storage is in advance popularity recently. In activity settings, we see the rise in demand for data outsourcing, which assists in the strategic management of corporate data. It is also used as a core technology behind many online services for personal applications. Now a days, it is easy to apply for free accounts for email, photo album, file sharing and/or remote access, with storage size more than 25GB (or a few dollars for more than 1TB). Together with the current wireless technology, users can access almost all of their files and emails by a mobile phone in any corner of the world. Considering data privacy, a traditional way to ensure it is to rely on the server to enforce the access control after authentication, which means any unexpected privilege escalation will expose all data.[1]

In a shared-tenancy cloud computing environment, things become even worse. Data from different clients can be hosted on separate virtual machines (VMs) but reside on a single physical machine.[2] Data in a target VM could be stolen by instantiating another VM co-resident with the target one. Regarding availability of files, there are a series of cryptographic schemes which go as far as allowing a third-party auditor to check the availability of files on behalf of the data owner without leaking anything about the data, or without Compromising the data owner's anonymity.[3],[4], Likewise, cloud users probably will not hold the strong belief that the cloud server is doing a good job in terms of confidentiality. A cryptographic solution, with proven security relied on number-theoretic assumptions is more desirable, whenever the user is not perfectly happy with trusting the security of the VM or the honesty of the technical staff.

A. Shakila Banu is with Department of Computer Science and Engineering, Chendhuran College of Engineering and Technology, Pudukkottai, Tamil Nadu, India. (Email : banusarthar@gmail.com)

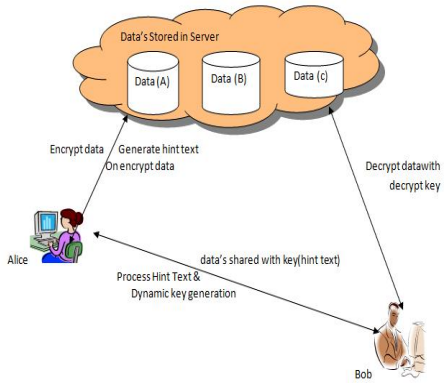

Fig.1. System Architecture

These users are motivated to encrypt their data with their own keys before uploading them to the server.

A. Data Sharing

Data's are shared in the cloud storage. For example, the users to share the resources to their friends and view a division of their secret images; and then the another user to access and responsive their data. This is one of the difficult problems to share their encrypted data in the cloud server and the another users can download their encrypted data from the cloud storage, finally the user decrypt the data's, and send them to sharing for other user's,[1].Users should be able to hand over the access rights of the sharing data to others so that they can access these data from the server directly. However, finding an able and secure way to share partial data in cloud storage is not unimportant. Assume that Alice puts all her private photos on Drop box, and she does not want to expose her photos to everyone. Due to various data outflow possibility Alice cannot feel relieved by just relying on the privacy protection mechanisms provided by Drop box, so she encrypts all the photos using her own keys before uploading. One day, Alice's friend, Bob, asks her to share the photos taken over all these years which Bob appeared in. Alice can then use the share function of Drop box, but the problem now is how to delegate the decryption rights for these photos to Bob. A possible option Alice can choose is to securely send Bob the secret keys involved..

B. Alice Encryption

Alice encrypts all files with a single encryption key and gives Bob the corresponding secret key directly. Alice encrypts files with distinct keys and sends Bob the corresponding secret keys. Obviously, the first method is inadequate since all un chosen data may be also leaked to Bob. For the second method, there are practical concerns on efficiency. The number of such keys is as many as the number of the shared photos, say, a thousand. Transferring these secret keys inherently requires a secure channel, and storing these keys requires rather expensive secure storage. The costs and complexities involved generally increase with the number of the decryption keys to be shared.

Encryption keys also come with two flavors — symmetric key or asymmetric (public) key. Using symmetric encryption, when Alice wants the data to be originated from a third party, she has to give the encrypt or her secret key; obviously, this is not always desirable. By contrast, the encryption key and decryption key are different in public-key encryption. The use of public-key encryption gives more flexibility for our applications.

For example, every employee can upload encrypted data on the cloud storage server without the knowledge of the company's master-secret key. Therefore, the best solution for the above problem is that Alice encrypts files with distinct public-keys, but only sends Bob a single (constant-size) decryption key. Since the decryption key should be sent via a secure channel and kept secret, small key size is always desirable. For example, we cannot expect large storage for decryption keys in the resource-constraint devices like smart phones, smart cards or wireless sensor nodes. Especially, these secret keys are usually stored in the tamper-proof memory, which is relatively expensive. The present research efforts mainly focus on minimizing the communication requirements (such as bandwidth, rounds of communication) like aggregate signature.

C. Sharing Encrypted data

A Key Aggregate Cryptosystem is an data sharing. But the key aggregation property is especially useful when we expect the allocation to be efficient and flexible. The schemes enable a content provider to share her data in a confidential and selective way, with a fixed and small cipher text expansion, by distributing to each authorized user a single and small aggregate key. Here we describe the main idea of data sharing in cloud storage using KAC, illustrated in Figure 2. Suppose Alice wants to share her data m1; m2; mon the server. She first performs Setup (1; n) to get param and execute Key Generation to get the public/master-secret key pair (pk; msk). The system parameter param and public-key pk can be made public and master-secret key msk should be kept secret by Alice. Anyone (including Alice herself) can then encrypt each mi by Ci = Encrypt (pk; i;mi). The encrypted data are uploaded to the server. With param and pk, people who cooperate with Alice can update Alice's data on the server. Once Alice is willing to share a set S of her data with a friend Bob, she can compute the aggregate key KS for Bob by performing Extract (msk; S). Since KS is just a constant size key, it is easy to be sent to Bob via a secure e-mail. After obtaining the aggregate key, Bob can download the data he is authorized to access.

II. ALGORITHM

```
dec arraylist (page_load, cmdsearch_click, DG_row command);
dec integer (j,k,l);
declare string (m,ws,a);
assign rowscount*(getrow cmdsearch_array()) to rowscount
assign row to cmdsearch
for j = 0 to rowscount - 1
cmdsearch.clear()
assign pageload(val  object,val event, " ") to Me.load;
 assign ubound(twords) to mfirst;
for j = 0 to ubound(twords)
 add twords(j) to first;
close loop[1];
  for k = 0 to oacalculation.nfreqset() - 1
check k is not equal to l, if so
second.clear();
assign split(freq_itmset1.item(l), " ") to twords;
nsecond = ubound(twords)
for l = 0 to ubound(twords)   add twords(l) to second;
   assign unionfs(first, mfirst, second, msecond) to unionstr;
assign assovalue(unionstr) to num;
assign assovalue(freq_itmset(j)) to dena;
      divide the value of num by dena and assign it to div1a;
      multiply div1a with 100;
      assign the result to confval;
      check confval is greater than or equal to confth, if so
      assign remove(freqitmset (j), freqitmset1 (k)) to status;
      check if the status is equal to 1, if so
      add num to anum;
      add dena to ardena;
         add(freqitmset (j)+freqitmset1 (k)) to assocrule;
      increment the assocount by 1;
   close condition[1];
   close condition[2];
   close condition[3];
   close loop[2];
   close loop[3];
for j = 0 to assocount - 1 then
disp1 = disp1 + associationrule(j) + vbcrlf
   obhiding.setasso(disp1);
   obhiding.setnasso(assocount);
   oitemset.itemsetinit();
   oacalculation.aprioriinit();
assoruleinit();
```

III. RELATED WORK

A. Existing Approach

In the existing system, the user shared the resources that are stored in the cloud. User send the encrypted file with set of

keys to the cloud, and then the both sender and receiver use the common keys to share resources.[6] The common keys to produce a result so, it is not secure to that resources. In the key aggregate crypto system using a multiple keys to produce set of results. And then the multiple keys are not remembered to that user in the existing system. So it is the drawbacks of these schemes.

B. Proposed System Approach

In the proposed system used a key aggregate crypto system,the aggregation is a collection of things to make a retrieve data but the key aggregate is a set of keys to produce a result. In this method the user share a resources use a multiple keys. The multiple keys are possible to limited resources but the user share a 'n' numbers of resources in the cloud, the keys are not remember to the user, so we can use the hint text methodology. The hint text methodology is easy to identify the key for a user share a resources. Then the keys are generated by user give the data's related so; it is secure for share the resources in the cloud storage. In additional the proposed system advantages of a cost balance, which is the user, share a resource at least possible to 3 documents and the 4^{th} document is uploaded use the cost balance.[9],[13].

C. Hint Text Manipulation

This system fully involves in the context of generating efficient hint texts against the given data. It is use un limited file because the user can't remember the keys so use the hint text, Once the user inputting the data this system asks the user to provide the hint text for manipulating the data against encryption, after that the hint text and the sampling data will be forwarded to the user's mail for clarification[7],[10].

D. Information Retrieval

The generic single database PIR protocol is built on a FHE scheme (KG, E, D, Add, Mult) and consists of three algorithms (Query Generation QG, Response Generation RG, and Response Retrieval RR). At a high level, the user generates a public and private key pair (pk; sk) for the FHE scheme, sends the public key pk to the database server, but keeps the private key sk secret. Then the user chooses an index i, where $1 < i < n$, and encrypts i with the public key pk, and sends the cipher text as a query to the database server. Based on the response generation circuit and homomorphic properties, the server computes an encryption of the ith bit as a response based on the database, the query and the public key pk, and sends the response back. At the end, the user decrypts the response to obtain the ith bit. Assume that the user and the database server have agreed upon a FHE scheme (KG, E, D, Add, Mult) in advance, our single-database PIR can be using Query generation, Response generation, Response Retrieval.[15]

E. Dynamic Decryption

In cryptography, Encryption is convert plain text into cipher text, and then the reverse process is called decryption. In the key aggregate crypto system produce a set of result using a advanced technique that is the method we use a onetime password. In this method the resources are shared in the cloud use the dynamic asymmetric key for decrypt the resources. Once the data sharing is completed then the key aggregate differs from its actual form. [7]The encrypted resources and the hint text is same but the decrypted hint text is a dynamic changes So the user cannot guess the key aggregate cryptosystem It is secure the user shared for the resources.[19]

F. Experimental Flow Diagram

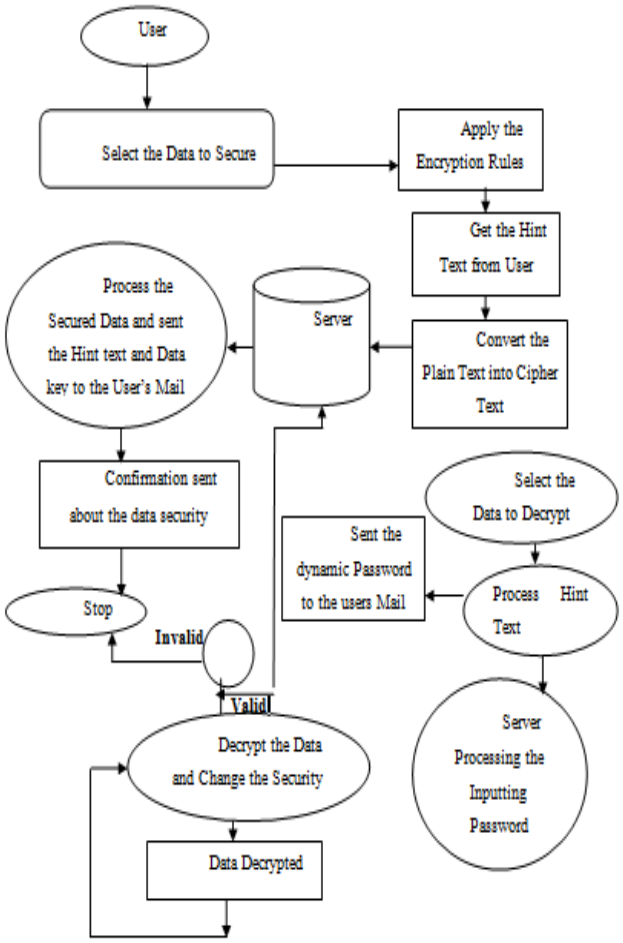

Fig.2. Experimental Flow Diagram

IV. ALGORITHM COMPARISION

The comparisons of two algorithm is important for find the SHA-with, Encrypt & Decrypt algorithm, Key generation algorithm that is User sends the public key to the database server, User chooses an index ,encrypts with public key , and sends the cipher text to the database server.

Key generation process is a multiple keys to produce a set of results. In advanced using a hint text manipulation it is dynamic generate the decrypt data for secure a resources in the cryptography.

V. CONCLUSION

In the key aggregate crypto system the multiple keys are produce a set of results. Then in which the multiple keys are

possible to limited resources but the user share the 'n' number of resources the keys are not remember to the user so we can use the hint text manipulation method .in this method very safe and secure to share the resources in the cloud storage.[15]

REFERENCES

[1] S. S. M. Chow, Y. J. He, L. C. K. Hui, and S.-M. Yiu, "SPICE -Simple Privacy-Preserving Identity-Management for Cloud Environment,"in Applied Cryptography and Network Security - ACNS2012, ser. LNCS, vol. 7341. Springer, 2012, pp. 526–543.

[2] L. Hardesty, "Secure computers aren't so secure," MIT press, 2009,http://www.physorg.com/news176107396.html.

[3] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Secure Cloud Storage," IEEE Trans.Computers, vol. 62, no. 2, pp. 362–375, 2013.

[4] B. Wang, S. S. M. Chow, M. Li, and H. Li, "Storing Shared Data on the Cloud via Security-Mediator," in International Conference on Distributed Computing Systems - ICDCS 2013. IEEE, 2013.

[5] S. S. M. Chow, C.-K. Chu, X. Huang, J. Zhou, and R. H. Deng,"Dynamic Secure Cloud Storage with Provenance," in Cryptographyand Security: From Theory to Applications - Essays Dedicated to Jean-Jacques Quisquater on the Occasion of His 65th Birthday, ser. LNCS, vol. 6805. Springer, 2012, pp. 442–464.

[6] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps," in Proceedings of Advances in Cryptology - EUROCRYPT '03, ser. LNCS, vol. 2656. Springer, 2003, pp. 416–432.

[7] M. J. Atallah, M. Blanton, N. Fazio, and K. B. Frikken, "Dynamic and Efficient Key Management for Access Hierarchies," ACM Transactions on Information and System Security (TISSEC), vol. 12, no. 3, 2009.