Attribute Based User Control Access for Cloud Storage Data

[1.]Satheesh Kavuri
Dhanekula Institute Of Engineering & Technology
Vijayawada, INDIA

[2.] GangadharaRao Kancharla
Dept. Of Computer Science & Engineering, ANU
Guntur, INDIA

[3.]Basaveswara Rao Bobba
Computer Center
AcharyaNagarjuaUniverisity
Guntur,INDIA

Abstract

Online documents provide very convenient ways for individuals to store and share various data including personal profile, business documents and etc on remote online data servers. Cloud Computing, regarded as the future IT architecture, and even promises to provide unlimited and elastic computing resources to cloud users in a very cost-effective way. Information security is basically a critical issue for remote data storage in cloud environment. On one side, disclosure of sensitive information, an example would be business records, stored on remote data servers ought to be strictly protected before users to utilize the data services. Traditional data access control mechanisms often should not set up appropriate security mechanism to sensitive data among multiple users. An untrusted storage cloud server is permitted to learn the content of sensitive data and doesn't enforce data access policies. To maintain data confidential owner need to encrypt data before upload to the cloud. User access is granted by possessing the data decryption secret key. In this proposed approach an improved attribute based document encryption and decryption is provided for multiple users in cloud environment.

***Keywords –Attribute based Encryption,* Cloud server, Amazon aws, Access policies.**

I.INTRODUCTION

Today's computing technologies have attracted larger numbers of people to accumulate their private data on third-party servers either for simplicity of sharing or cost saving. When people enjoy the advantages these new technologies and services cause, their concerns about data security also arise. Naturally, people wish to make their private data only accessible to authorized users. In many cases, additionally it is desirable to provide differentiated access services such that data access policies are defined over user attributes/roles. It's pretty easy to foresee that these security concerns and requirements would become more urgent among the coming era of the cloud wherein individuals, organizations, and businesses may outsource their various types files, also the highly sensitive data, directly into cloud.

Cipher text-policy attribute based encryption (CP-ABE) [2] serves as a public-key cryptography primitive that had been proposed to resolve the main concern about fine-grained access control on shared data in one-to-many communications. In CP-ABE, each user is assigned specific attributes which are embedded straight into the user's secret key. A public key component is defined for each individual user attribute. When encrypting the message, the encryptor chooses an access structure on attributes, and encrypts what it s all about below the access structure via encrypting with the corresponding public key components. Users are willing to decrypt a ciphertext if and only if their attributes satisfy the ciphertext access structure. The end users key and ciphertext sizes in CP-ABE are only linear into the wide range of attributes and of course the complexity of one's access structure, which happens to be independent to the number of users. Moreover, CP-ABE is proof against collusion attacks from unauthorized users. Each one of these nice properties make CP-ABE extremely well suited for fine-grained data control access on untrusted storage.

Ciphertext-Policy Attribute-Based Encryption (CP-ABE) enables us to encrypt data under an access policy, specified as a Boolean formula over attributes. Ciphertexts are only able to be decrypted by users who possess all attributes required to satisfy the access policy. Decryption is performed by making use of secret attribute keys; one such attribute key corresponds to one attribute of an individual. Thus, only a user who possesses the right number and mixture of these secret keys are able to access the data. Connected to most previous ABE schemes would be the occurrence of a central trusted authority (master) that knows a secret master key and distributes attribute keys to eligible users. However, for a lot of practical scenarios that take pleasure in the use of attribute-based encryption, there is virtually no central authority that is undoubtedly in a position to maintain all attributes and distribute secret attribute keys[3-5].

To encrypt information, a user first formulates his access policy. Dependent upon the construction the sort of this policy might be a Boolean formula, a linear secret sharing scheme, or perhaps a different formalism. The buyer can finally encrypt messages within a policy by

utilizing the public keys corresponding to the attributes occurring among the policy. To decrypt a ciphertext, an individual needs not less than access to some set of attributes (and their associated attribute keys) which satisfies the access policy. If he doesn't already possess these keys, he may query the attribute authorities for your secret keys akin to the attributes he is eligible for[1].
Public-Key encryption is a powerful mechanism for safeguarding the confidentiality of stored and transmitted information. Traditionally, encryption is viewed as a strategy to acquire a user to present data to some targeted user or device. Even though this is useful for applications exactly where the data provider knows specically which user he wants to show, in lots of applications the provider will want to share data in accordance with some policy driven by receiving user's credentials[6-8].

II.LITERATURE SURVEY

IBE, an identity or ID can be an string 1-1 mapped to each and every user. An individual can acquire a unique key equivalent to his/her ID inside an online manner from trusted authority plus the ID is made use of as public key. The ciphertext encrypted by the particular ID are only able to be decrypted by the user with corresponding private key, i.e., the encryption is one-to-one[2].

In the traditional CP-ABE scheme, once users obtain the credentials typically from a system manager at the start of setup phase, the access ability is often valid for individuals who may even break the confidential rules by abusing these financial private information. Upon detecting those malicious adversaries, with no revocation mechanism embedded, the internal system manager is required to rebuild in the whole system. Therefore, revocation mechanism should be designed into your system immediately rather than just being added later on different problems are addressed, mainly because it requires careful planning on where functionality should really be placed and the best way to reduce the computational and communication costs[3,4].

Recently, much attention has also been attracted by a new public key primitive called Attribute-based encryption (ABE). ABE has significant advantage during the traditional PKC primitives mainly because it achieves flexible one-to-many encryption instead of one-to-one. ABE is envisioned as a possible important tool for addressing the trouble of secure and fine-grained data sharing and access control. Inside an ABE system, a practitioner is identified by way of a variety of attributes. A secret key based on a set of attributes ω, can decrypt a ciphertext encrypted utilizing a public key based on particular attributes ω', only if the sets ω and ω' overlap sufficiently as influenced by a threshold value t. A user could encrypt a document to the shared users who have certain range of attributes drawn typically from a pre-defined attribute universe.

A user identity (like name, e-mail address and so on) may be used for accessing influence over some resources. For example, in Identity-Based Encryption (IBE) schemes , an encryptor can restrict a decryptor to indicate the identity of a given decryptor. An Attribute-Based Encryption (ABE) can be an encryption scheme, where users with many attributes can decrypt the ciphertext involved with these attributes. Although IBE schemes possess a restriction so that an encryptor only indicates a single decryptor, in ABE schemes, an encryptor can indicate many decryptors by assigning common attributes for these decryptors an example would be gender, age, affiliation and many more[9].

Selective Data Sharing

Look at a scenario in which a large corporation installs a standing committee to research any reports of improper conducts of employees. Participants in this committee are drawn from different departments and locations, and therefore are given three different clearance levels. Using a hierarchical CP-ABE instance identical to that in Section 5, these attributes can easily be categorized like for example as shown in Fig 1.

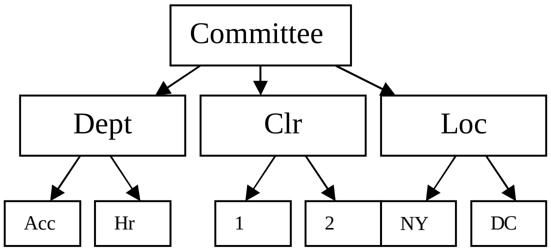


Fig 1: Attributes Category

Suppose there is certainly a study regarding an accounting officer in New York, and company policy says that no committee member that are caused by the accounting department in Big apple able to sit for aspect in this investigation. To encrypt a memo regarding this investigation, your content of the memo is first encrypted with the use of a symmetric data key. Information key is then encrypted separately with the AND gates in Figures. The two ciphertexts are placed within the header that accompanies the encrypted memo. Anyone not belonging to an accounting department can decrypt the first ciphertext, and anyone isn t working in Big apple can decrypt the 2nd. This enforces the need access policy[10].

III. PROPOSED SYSTEM

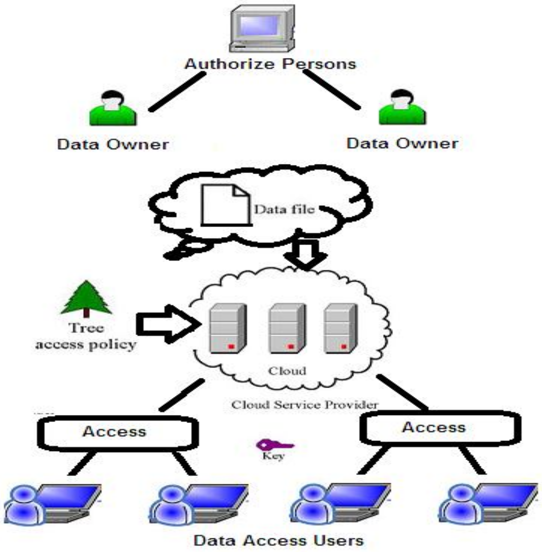


Fig. 2 Overview of Proposed Work

In the above architecture diagram Fig 2, Authorized persons are documents owners who can share the documents in the cloud to authorized users in the attribute list. Each document has one or more data owners who have permission to access the cloud. Data owners secure the data files in the cloud with access policy tree and cloud access permissions. In this process each file is tokenized to reduce the size of the document during encryption and decryption process. All data access users will get the secret key which is used to decrypt the cloud document with access tree policy structure.

The description of these algorithms is as follows:
Setup: The *Setup* algorithm takes as input the security parameter δ . It outputs the public key PK which is used in all subsequent algorithms and the secret master key MK as shown in Fig. 3.

Cloud Policy Based Attribute-based encryption:

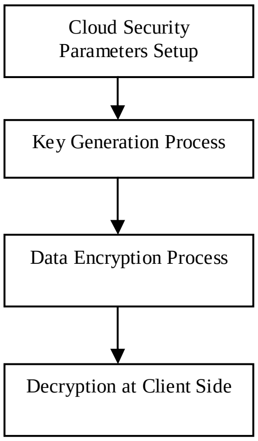


Fig. 3: Attribute Based Cloud Security

Setup: This algorithm takes as input a security parameter κ and returns the public key PK and a system master secret key MK. For encryption message senders uses the PK. User secret keys generated by using MK and are known only to the authority.
Key Generation: This algorithm takes access structure T and the master secret key MK as input and outputs SK secret key T.
Encryption: This algorithm takes as input a message M, the public key PK, and a set o attributes .It outputs the cipher text E.
Decryption: User's secret key is generated from the cloud storage and then user verifies his access structure T. User takes cipher text as input and then produces message M as output.

SETUP PROCESS:

Let u = $\{ap1,ap2.....\ ap_k\}$ be the *Set* of authorized attributes profiles in the system. Each at_i has three values: at_i^+ denotes the user possess at_i , at_i^- denotes the user does not have at_i , at_i^* denotes user has either at_i^+ or at_i^- [4] as in Fig 4.

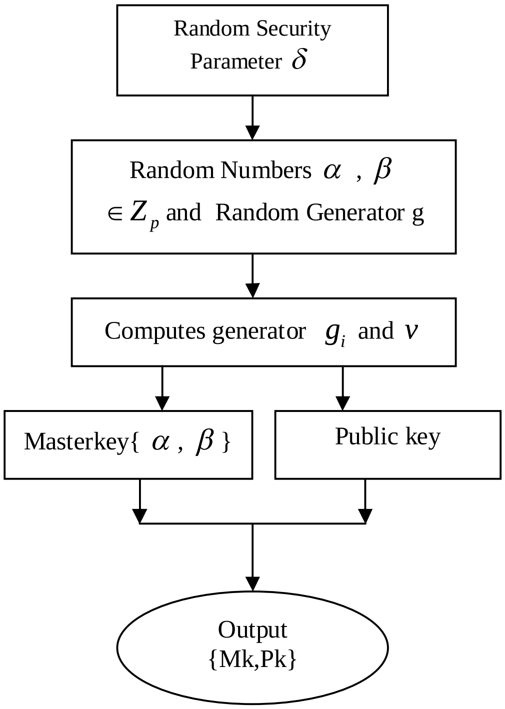

Random Security Parameter δ
Random Numbers α , β $\in Z_p$ and Random Generator g
Computes generator g_i and v
Masterkey{ α , β }
Public key
Output {Mk,Pk}

Fig.4 . SetUp Execution Process

Bilinear Maps:

Pairing uses bilinear map function E: $MC_0 \times MC_0 - > MC_1$, where MC0 and MC1 are two multiplicative cyclic groups with p large prime order.

Properties:

Bilinearity:

$$E(X^p, Y^q) = E(X, Y)^{pq}, \forall X, Y \in MC_0, \forall p, q \in Z_p^*$$

Nondegeneracy:

E(g,g) $\neq 1$ where g is generator of C_0

In this process user selects security related document and security parameter δ and produce output public key as well as master key.

KEY GENERATION:

The algorithm proceeds as shown below. Each user, pick a random polynomial qx for any non-leaf node x inside the universal access tree T'[7]. These polynomials are chosen in the following means by a top-down manner, ranging from the main node r. Each x, set the degree cx of a given polynomial qx to be one less than the edge value, i.e., cx = num− 1. Now, for your root node r, set qr(0) = y and choose cr other points of one's polynomial qr randomly to reflect it completely. For every other non-leaf node x, set qx(0) = qparent(x) (index(x)) and choose cx other points randomly to completely define qx. When the polynomials have already been decided, provide following secret values towards the user:

$$D_{j,x} = g^{q_x(j)/t_{j,x}}\} j \in \phi, x \in \chi(T^1)$$

ENCRYPTION AND DECRYPTION PROCESS:

The encryption algorithm encrypts a message M under the tree access structure T' . The algorithm first chooses a polynomial qx for any node x (that includes the leaves) within the tree T' . These polynomials are chosen in the following way in a topdown manner, ranging from the root node R. Each node x inside the tree, set the degree dx of one's polynomial qx to become one below the string value kx of the node, that really is, dx = kx − 1. The ciphertext is then constructed by giving the tree access structure T'. Reverse process of encryption is decryption process. Using secret key and cipher text original message is generated.

IV. EXPERIMENTAL RESULTS

All experiments are performed with the configurations Intel(R) Core(TM)2 CPU 2.13GHz, 2 GB RAM, and the operating system platform is Microsoft Windows XP Professional (SP2). This framework requires third party libraries abe,cpabe.

Cloud Instance results :[{ReservationId: r-3393c53a,OwnerId: 751633946423,Groups: [],GroupNames: [],Instances: [{InstanceId: i-b11ffcba,ImageId: ami-1e3a502e,State: {Code: 80,Name: stopped},PrivateDnsName: ip-172-31-47-124.us-west-2.compute.internal,PublicDnsName: ,StateTransitionReason: User initiated (2014-05-04 05:41:00 GMT),KeyName: kavuri,AmiLaunchIndex: 0,ProductCodes: [],InstanceType: t1.micro,LaunchTime: Sun May 04 08:09:10 IST 2014,Placement: {AvailabilityZone: us-west-2b,GroupName: ,Tenancy: default},KernelId: aki-f08f11c0,Monitoring: {State: disabled},SubnetId: subnet-babcaed8,VpcId: vpc-805d4fe2,PrivateIpAddress: 172.31.47.124,StateReason: {Code: Client.UserInitiatedShutdown,Message: Client.UserInitiatedShutdown: User initiated shutdown},Architecture: i386,RootDeviceType: ebs,RootDeviceName: /dev/sda1,BlockDeviceMappings: [{DeviceName: /dev/sda1,Ebs: {VolumeId: vol-c9bc4ec8,Status: attached,AttachTime: Sun May 04 08:09:13 IST 2014,DeleteOnTermination: true}}],VirtualizationType: paravirtual,ClientToken: oXoXK1399171149482,Tags: [],SecurityGroups: [{GroupName: launch-wizard-7,GroupId: sg-f18d5e94}],SourceDestCheck: true,Hypervisor: xen,NetworkInterfaces: [{NetworkInterfaceId: eni-f7068f92,SubnetId: subnet-babcaed8,VpcId: vpc-805d4fe2,Description: ,OwnerId: 751633946423,Status: in-use,PrivateIpAddress: 172.31.47.124,PrivateDnsName: ip-172-31-47-124.us-west-2.compute.internal,SourceDestCheck: true,Groups: [{GroupName: launch-wizard-7,GroupId: sg-f18d5e94}],Attachment: {AttachmentId: eni-attach-1398d824,DeviceIndex: 0,Status: attached,AttachTime: Sun May 04 08:09:10 IST 2014,DeleteOnTermination: true},PrivateIpAddresses: [{PrivateIpAddress: 172.31.47.124,PrivateDnsName: ip-172-31-47-124.us-west-2.compute.internal,Primary: true,}]}],EbsOptimized: false}]}]

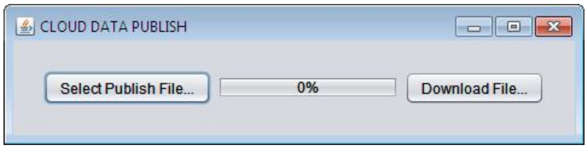
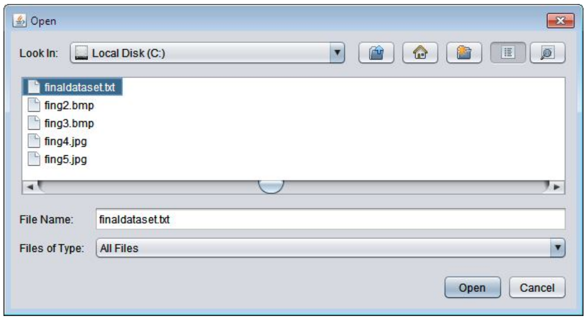


Fig. 5: Cloud Data Publishing Screen



Look In: Local Disk (C:)

finaldataset.txt
fing2.bmp
fing3.bmp
fing4.jpg
fing5.jpg

File Name: finaldataset.txt

Files of Type: All Files

Fig. 6: Select data to publish

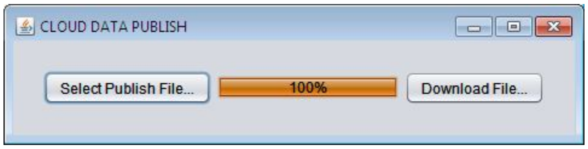
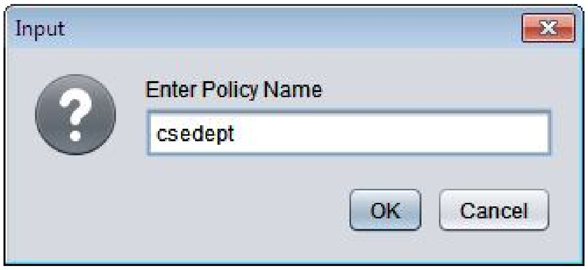


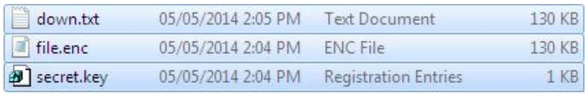
Fig. 7: Data Published in Cloud



Enter Policy Name

csedept

Fig. 8: Download Published Data



down.txt	05/05/2014 2:05 PM	Text Document	130 KB
file.enc	05/05/2014 2:04 PM	ENC File	130 KB
secret.key	05/05/2014 2:04 PM	Registration Entries	1 KB

Fig.9 : Downloaded Files from Cloud

Performance Analysis

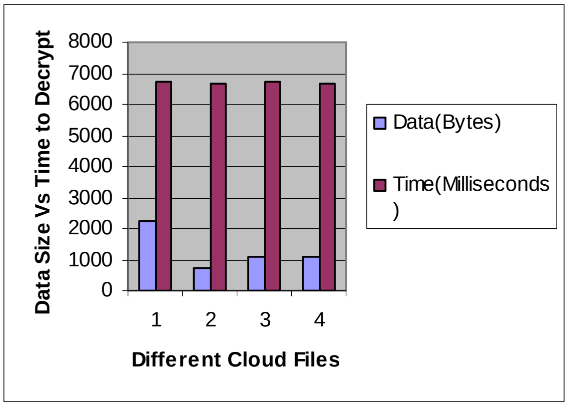


Fig.10 : Comparison between data data sizes with Time to decrypt from cloud

V. CONCLUSION AND FUTURE SCOPE

Our approach is against password type of attacks which are prevalent against the existing Schemes. Experimental results shows cloud data security is verified successfully before accessing the document. Each user is authenticated using access tree structure as user policy. This approach has some limitations 1) if the size of the cloud document increases then it becomes very difficult to encrypt the data using existing ABE method. 2) Access tree structure size increases as number of policies per user increases. 3) Doesn't supportable to other data file formats. We can extend the existing scheme for other attributes like location, time etc. Such a scheme would employ new access policy scheme compare to existing tree based approach.

VI. REFERENCES

[1]. Ciphertext-Policy Attribute-Based Encryption John Bethencourt, Security and Privacy, 2007. SP '07. IEEE Symposium, 321 – 334,2007.

[2] Attribute Based Data Sharing with Attribute Revocation Shucheng Yu, Cong Wang. Proceeding ASIACCS '10 Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security
Pages 261-270 ACM New York, NY, USA ©2010.

[3] A Survey on Attribute-based Encryption Schemes of Access Control in Cloud Environments Cheng-Chi Lee , Pei-Shan Chung , and Min-Shiang Hwang, International Journal of Network Security, Vol.15, No.4, PP.231-240, July 2013.

[4] Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption, Ming Li, Jan. 2013 (vol. 24 no. 1) pp. 131-143.

[5] Security of PHR in Cloud Computing by Using Several Attribute Based Encryption Techniques   Neetha Xavier, V.Chandrasekar, International Journal of Communication and Computer Technologies Volume 01 – No.72 Issue: 07 Nov 2013 ISSN NUMBER : 2278-9723 .

[6] Scalable Access Control in Cloud Computing Using
Hierarchical Attribute Set Based Encryption (HASBE) A.Vishnukumar, G.Muruga Boopathi, S.Sabareessh, International Journal of Emerging Science and Engineering (IJESE) ISSN: 2319–6378, Volume-1, Issue-4, February 2013

[7] Rakesh Bobba, Himanshu Khurana and Manoj Prabhakaran, "AttributeSets: A Practically Motivated Enhancement to Attribute-Based Encryption", July 27, 2009

[8] R. Ostrovsky and B. Waters. "Attribute based encryption with nonmonotonic access structures".In Proceedings of the 14th ACM conference on Computer and communications security, pages 195{203. ACM New
York, NY, USA,2007.

[9] K. Emura, A. Miyaji, A. Nomura, K. Omote, and M. Soshi, \A ciphertext-policy attribute-based encryption scheme with constant ciphertext length," in Proceedings of the Information Security Practice and Experience, pp. 13{23, 2009.

[10] V. Goyal, A. Jain, O. Pandey, and A. Sahai, \Bounded ciphertext policy attribute based encryption," in Proceedings of the ICALP, pp. 579{591, 2008.