HIERARCHICAL LEVEL SECURITY IN CLOUD COMPUTING

Vikas Mahapatra[1], Shreyas Khanadagale[2], Anusen Bale[3] , Prof Pallavi Chandratre[4]

Department of Computer Engineering, Shivajirao S. Jondhale College of Engineering, Dombivli Maharashtra, India.

---***---

Abstract - *In our paper, we increase the security on attribute based solution concepts and provide additional security for HASBE. This scheme is suggested for access governor in cloud computing. To achieve springy, accessible and fine-grain access control HASBE prolongs the cipher text policy attribute based encryption scheme.*

Key Words: Attribute based encryption, cloud computing data security, cipher text.

1. INTRODUCTION

We use the Hierarchical structure for registration i.e. first the Domain Authority on the cloud which will provide a unique id for the user. This unique id will be used for the user registration. After the successful login of the user, the user will be provided with two options, Data owner and Data Consumer. Since the user has not uploaded anything on the cloud so the user is not a data owner. After the user uploads the file in the cloud, master key is generated through attributes of that files. This master key is generated by bilinear mapping i.e. by using AND gate over some attributes.

After the generation of the master key, the Secret Key is generated by applying OR gate over Public key (obtained by binary format of the user name) and Master key. For Encryption and Decryption of data Secret key is used. We use VMware for the storage of data. When the user becomes Data Consumer from Data Owner, the user need Secret Key to retrieve the data from the cloud. We prove that how HASBE system prolongs the ASB-Encryption algorithm with Hierarchical structure. We have shown that how access control is is done on cloud computing through HASBE. File creation, file deletion in cloud computing is support through this scheme. Our project demonstrate that HASBE has good performance

1.1 PROBLEM DEFINITION

The existing system applies cryptographic methods by disclosing data encryption keys only to authorize users. For computing on the data owner for key distribution and data management, these solutions introduce a heavy load

Many numbers of schemes have been proposed, for achieving flexible and fine-grained access control. Unfortunately, in our system the data holder and end user are one and the same. Since data holder and service suppliers are not in the same trusted domain in cloud, to over come that new access control scheme employing AB(Attribute Based) encryption is needed.

2. RESEARCH AND RELATED WORK

In the paper [1], for creating clouds architecture is provided and the cloud is defined using technologies such as Virtual Machines.

We recognize a algorithm for effective two-policy reconciliation, and show that, in the worst-case, reconciliation of more than three policies is inflexible [2]. Now, we suggest effective heuristics for the finding and resolution of intractable reconciliation. Depending on the policy model, we define the design and implementation of the policy language.

In this paper [3], we present a scheme for realizing multifaceted access control on encrypted data that we call Ciphertext-Policy Attribute-Based Encryption. By using our techniques encrypted data can be kept confidential even if the storage server is untrusted; moreover, our methods are secure against collusion attacks. Back in time, ABE system was using attribute for defining the encrypted information (or data) and also to develop policies into user keys; whereas in our system, attributes are used for describing a user's identification, and a party encrypting data defines a policy for who can decrypt.

A Fuzzy IBE [4] scheme can be applied to allow encryption using biometric measurements as identities. The error-tolerance of the Fuzzy IBE scheme is exactly what allows for the usage of biometric identities, which inherently comprise some amount of noise during each measurement.

3. PROPOSED SYSTEM

We propose the HASBE: Hierarchical Attribute Set-Based Solution in Cloud Computing. HASBE extends the cipher text-policy attribute- set-based encryption (CP-ASBE) system with the hierarchical structure of the system user, so they can accomplish scalable, flexible and also fine-grained access control. Specifically, we assign each and every data file with a set of attributes and allocate each and every user an expressive access structure which is defined above these attribute.

Data privacy is also attained because Cloud Servers cannot learn the plaintext of any of the data file in our system.

The following represents the designs of our proposed system:-

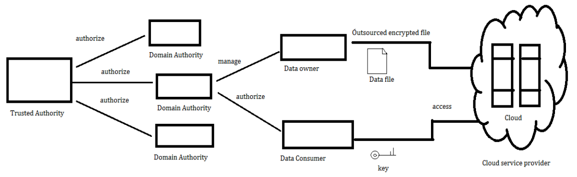


Fig- 1: Architectural design

We are using the following four modules in this project:

- Data Holder.
- Data End user.
- Cloud Server.
- Attribute centered key creation.
- Data Holder.

In this phase, the data holder uploads the file in the cloud server. To make more secure, the data owner encrypts the file and then store in the cloud. The data owner has the ability to change the expiry time of the files and also manipulate the encrypted file. It can set the access right to the encrypted file.

- Data End user.

In this module, the user has right to access the data file using the encrypted key. all the rights are given by domain authority to the user. User try to access the file within or outside the scope of the access right

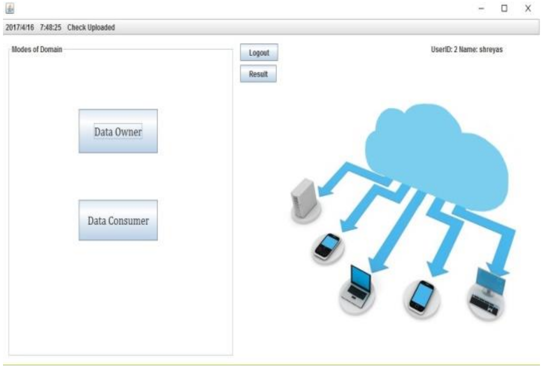
- Cloud Server.

In this module, it provides a data storage service. The file used for sharing between the data owner and customers is stored in the cloud server. The data consumers can download the encrypted the file from the cloud server as per their need and after downloading then decrypt it

- Attribute centered key creation.

In this module, trusted authority manager all the keys and parameters as well as authorizating domain authority. The domain authority is responsible for distributing key to next level of authorities. Each user in the system has a key which define the user's decryption key. To create Public Key (P.K) & Master Key (M.K), the trusted authority call the algorithms. The secret key are kept secret whereas Public Key (P.K) are made public when the user request for the file which is stored on the cloud. The cloud server sends the corresponding cipher text of that file to the user, then the user decrypts it with the decrypting key.

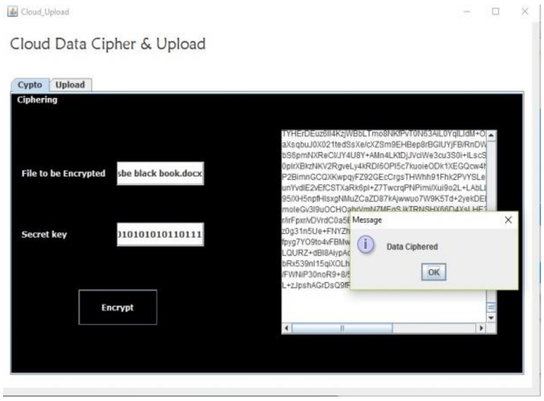
4. RESULTS


2017/4/16 7:48:25 Check Uploaded

Modes of Domain

UserID: 2 Name: shreyas

Fig- 2: Modes of Domain


Cloud_Upload

Cloud Data Cipher & Upload

Ciphering

File to be Encrypted: sbe black book.docx

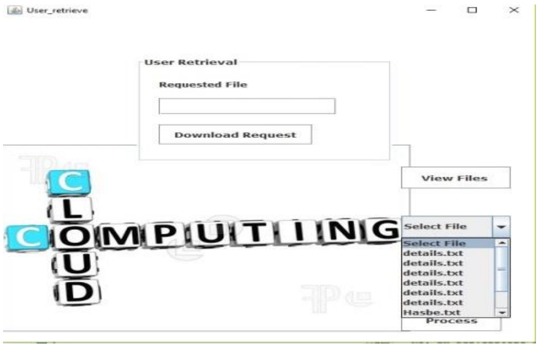
Secret key:)10101010110111

```
TYHErDEuz6ll4KzjWBbLTmo8NKfPvT0N63AiL0YqlLldM+O
aXsqbuJ0X021tedSsXe/cXZSm9EHBep8rBGlUYjFB/RnDW
bS6pmNXReCl/JY4U8Y+AMn4LKtDjJVciWe3cu3S0i+ILscS
0plrXBkzNKV2RgveLy4kRDI6OPI5c7kuoieODk1XEGQcw4
P2BimnGCQXKwpqyFZ92GEcCrgsTHWhh91Fhk2PVYSLe
unYvdlE2vEfCSTXaRk6pl+Z7TwcrqPNPimi/Xui9o2L+LAbLl
95/XH5npfHlsxgNMuZCaZD87kAjwwuo7W9K5Td+2yekDE
moleGv3l9uOCHO
r/irFpxrMDVrdC0a5E
z0g31n5Ue+FNYZh
fpyg7YO9to4vFBMw
LQURZ+dBI8AlypAc
bRx539nl15qiXOLh
/FWNiP30noR9+8/5
L+zJpshAGrDsQ9fF
```

Message

Data Ciphered

Fig- 3: File Encryption

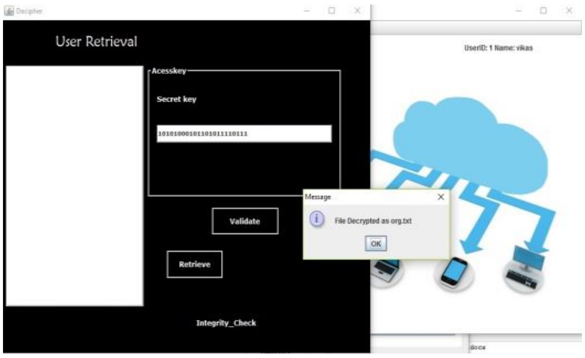

User_retrieve

User Retrieval

Requested File

Select File
details.txt
details.txt
details.txt
details.txt
details.txt
details.txt
Hasbe.txt

Fig- 4: File Requesting


Decipher

User Retrieval

Acesskey

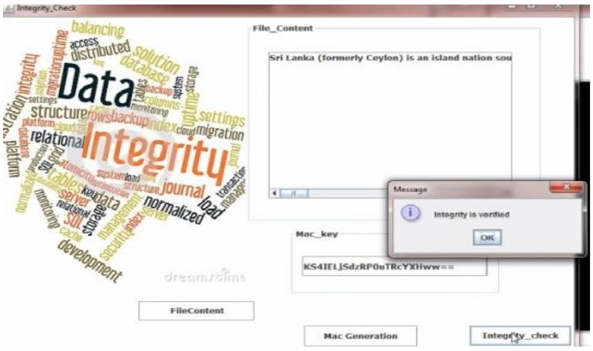
Secret key: 1010100010110101 1110111

UserID: 1 Name: vikas

Message

File Decrypted as org.txt

Fig- 5: Decryption of file



Integrity_Check

File_Content: Sri Lanka (formerly Ceylon) is an island nation sou

Message: Integrity is verified

Mac_key: KS4IELjSdzRP0uTRcYXHww==

Fig- 6: Integrity check

4. CONCLUSIONS

Thus we have implemented Hierarchical Level Security In Cloud Computing by providing a scalable, flexible and fine grained access to the cloud users. The Hasbe scheme seamlessly incorporates a hierarchical structure of system users by applying a delegation algorithm to ASBE.

REFERENCES

[1] R. Buyya, C. ShinYeo, J. Broberg, and I. Brandic, “Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility,” Future Generation Comput. Syst., vol. 25, pp. 599–616, 2009.

[2] P. D. McDaniel and A. Prakash, “Methods and limitations of security policy reconciliation,” in Proc. IEEE Symp. Security and Privacy, Berkeley, CA, 2002.

[3] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-policy attribute based encryption,” in Proc. IEEE Symp. Security and Privacy, Oakland, CA, 2007.

[4] A. Sahai and B. Waters, “Fuzzy identity based encryption,” in Proc. Advances in Cryptology—Euro crypt, 2005, vol. 3494, LNCS, pp. 457–473.