Explore HASBE Scheme for Fine- Grained Access Control of Outsourced Data in Cloud Computing

Dhanya V Kurup

ABSTRACT-- Cloud storage enables users to remotely store their data and use the on-demand high quality cloud applications without the burden of local hardware and software management. Though the benefits are clear, such a service is also relinquishing users' physical possession of their outsourced data, which inevitably poses new security risks towards the correctness of the data in cloud. In order to address this new problem and further achieve a secure and dependable cloud storage service, we propose in this project a flexible distributed storage integrity auditing mechanism, utilizing the homomorphism token and distributed erasure-coded data. The proposed design allows users to audit the cloud storage with very lightweight communication and computation cost. The auditing result not only ensures strong cloud storage correctness guarantee, but also simultaneously achieves fast data error localization, i.e., the identification of misbehaving server. Considering the cloud data are dynamic in nature, the proposed design further supports secure and efficient dynamic operations on outsourced data, including block modification, deletion, and append. Analysis shows the proposed scheme is highly efficient and resilient against Byzantine failure, malicious data modification attack, and even server colluding attacks.

Index Terms— ASBE,CP-ABE,DABE,DEK,HASBE,HIBE,IBE

IJSER

I. INTRODUCTION

Several trends are opening up the era of Cloud Computing, which is an Internet-based development and use of computer technology. The ever cheaper and more powerful processors, together with the software as a service (SaaS) computing architecture, are transforming data centers into pools of computing service on a huge scale. The increasing network bandwidth and reliable yet flexible network connections make it even possible that users can now subscribe high quality services from data and software that reside solely on remote data centers. Moving data into the cloud offers great convenience to users since they don't have to care about the complexities of direct hardware management.

The pioneer of Cloud Computing vendors, Amazon Simple Storage Service (S3) and Amazon Elastic Compute Cloud (EC2) are both well-known examples.

- *Dhanya V.Kurup received M.E degree in Computer Science and Engineering from Anna University Chennai, India. She is a lecturer in AcharyaInstituteof Technology,India. Email:dhanyakurup@acharya.ac.in*

amounts of storage space and customizable computing resources, this computing platform shift, however, is eliminating the responsibility of local machines for data maintenance at the same time.

As a result, users are at the mercy of their cloud service providers for the availability and integrity of their data . On the one hand, although the cloud infrastructures are much more powerful and reliable than personal computing devices, broad range of both internal and external threats for data integrity still exist. Examples of outages and data loss incidents of noteworthy cloud storage services appear from time to time . On the other hand, since users may not retain a local copy of outsourced data, there exist various incentives for cloud service providers (CSP) to behave unfaithfully towards the cloud users regarding the status of their outsourced data. For example, to increase the profit margin by reducing cost, it is possible for CSP to discard rarely accessed data without being detected in a timely fashion. Similarly, CSP may even attempt to hide data loss incidents so as to maintain a reputation.

Therefore, although outsourcing data into the cloud is economically attractive for the cost and complexity of long-term large-scale data storage, its lacking of offering strong assurance of data integrity and availability may impede its wide adoption by both enterprise and individual cloud users. In order to achieve the assurances of cloud data integrity and availability and enforce the quality of cloud storage service, efficient methods that enable on-demand data correctness verification on behalf of cloud users have to be designed. However, the fact that users no longer have physical possession of data in the cloud prohibits the direct adoption of traditional cryptographic primitives for the purpose of data integrity protection. Hence, the verification of cloud storage correctness must be conducted without explicit knowledge of the whole data files. Meanwhile, cloud storage is not just a third party data warehouse. The data stored in the cloud may not only be accessed but also be frequently updated by the users, including insertion, deletion, modification, appending, etc. Thus, it is also imperative to support the integration of this dynamic feature into cloud storage correctness assurance, which makes the system design even more challenging. Last but not the least, the deployment of Cloud Computing is powered by data centers running in a simultaneous, cooperated and distributed manner. It is more advantages for individual users to store their data redundantly across multiple physical servers so as to reduce the data integrity and availability threats.

As a complementary approach, researchers have also proposed distributed protocols for ensuring storage correctness across multiple servers or peers. However, while providing efficient cross server storage verification and data availability insurance, these schemes are all focusing on static or archival data. As a result, their capabilities of handling dynamic data remains unclear, which inevitably limits their full applicability in cloud storage scenarios. In this paper, we propose an effective and flexible distributed storage verification scheme with explicit dynamic data support to ensure the correctness and availability of users' data in the cloud. We rely on erasure correcting code in the file distribution preparation to provide redundancies and guarantee the data dependability against Byzantine servers , where a storage server may fail in arbitrary ways. This construction drastically reduces the communication and storage overhead as compared to the traditional replication-based file distribution techniques.

II. OBJECTIVE AND SCOPE OF WORK

In the existing system they propose hierarchical attribute-set-based encryption (HASBE) by extending ciphertext-policy attribute-set-based encryption (ASBE) with a hierarchical structure of users. The proposed scheme not only achieves scalability due to its hierarchical structure, but also inherits flexibility and fine-grained access control in supporting compound attributes of ASBE. In addition, HASBE employs multiple value assignments for access expiration time to deal with user revocation more efficiently than existing schemes. We formally prove the security of HASBE based on security of the cipher text-policy attribute-based encryption (CP-ABE) scheme by Bethen court et al. and analyze its performance and computational complexity. Implement our scheme and show that it is both efficient and flexible in dealing with access control for outsourced data in cloud computing with comprehensive experiments.

More specifically, we associate each data file with a set of attributes, and assign each user an expressive access structure which is defined over these attributes. To enforce this kind of access control, we utilize KP-ABE to escort data encryption keys of data files. Such construction enables us to immediately enjoy fine-grained of access control. However, this construction, if deployed alone, would introduce heavy computation overhead and cumbersome online burden towards the data owner, as he is in charge of all the operations of data/user management. Specifically, such an issue is mainly caused by the operation of user revocation, which inevitably requires the data owner to re-encrypt all the data files accessible to the leaving user, or even needs the data owner to stay online to update secret keys for users. To resolve this challenging issue and make the construction suitable for cloud computing, we uniquely combine PRE with KP-ABE and enable the data owner to delegate most of the computation intensive operations to Cloud Servers without disclosing the underlying file contents. Such a construction allows the data owner to control access of his data files with a minimal overhead in terms of computation effort and online time, and thus fits well into the cloud environment. Data confidentiality is also achieved since Cloud Servers are not able to learn the plaintext of any data file in our construction. For further reducing the computation overhead on Cloud Servers and thus saving the data owner's investment, we take advantage of the lazy re-encryption technique and allow Cloud Servers to "aggregate "computation tasks of multiple system operations. As we will discuss in section V-B, the computation complexity on Cloud Servers is either proportional to the number of system attributes, or linear to the size of the user access structure/tree, which is independent to the number of users in the system. Scalability is thus achieved. In addition, our construction also protects user access privilege information against Cloud Servers. Accountability of user secret key can also be achieved by using an enhanced scheme of KP-ABE. call that our system model consists of a trusted authority, multiple domain authorities, and numerous users corresponding to data owners and data consumers. The trusted authority is responsible for generating and distributing system parameters and root master keys as well as authorizing the top-level domain authorities.

III. SYSTEM MODEL AND ASSUMPTIONS.

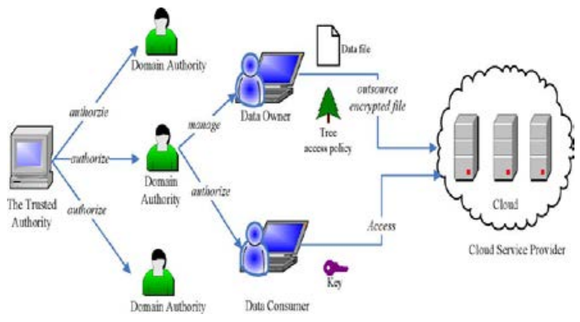

Fig .1 System model

A: System Model

As depicted in Fig. 1, the cloud computing system under consideration consists of five types of parties: a *cloud service provider*, *data owners*, *data consumers*, a number of *domain authorities*, and a *trusted authority*.
The cloud service provider manages a cloud to provide data storage service. Data owners encrypt their data files and store them in the cloud for sharing with data consumers. To access the shared data files, data consumers download encrypted data files of their interest from the cloud and then decrypt them. Each data owner/consumer is administrated by a domain authority. A domain authority is managed by its parent domain authority or the trusted authority. Data owners, data consumers, domain authorities, and the trusted authority are organized in a hierarchical manner as shown in Fig. 1. The trusted authority is the root authority and responsible for managing top-level domain authorities. Each top-level domain authority corresponds to a top-level organization, such as a federated enterprise, while each lower-level domain authority corresponds to a lower-level organization, such as an affiliated
company in a federated enterprise. Data owners/consumers may correspond to employees in an organization. Each domain authority
is responsible for managing the domain authorities at the next level or the data owners/consumers in its domain. In our system, neither data owners nor data consumers will
be always online. They come online only when necessary, while
the cloud service provider, the trusted authority, and domain authorities are always online. The cloud is assumed to have abundant storage capacity and computation power. In addition, we assume that data consumers can access data files for reading only.

***B*: Root and Domain Authority Registry**

Root authority holds the top most priority in the secure cloud storage and access system and he administrates the domain authority. Root authority has to register first in the cloud in order to get the services and to manage all the resources in the cloud.

Next to root authority, Domain authority administrates the data owner who owns the data in the cloud.

***C*: Bilinear Mapping**

In mathematics, a bilinear operator is a function combining elements of two vector spaces to yield an element of a third vector space that is linear in each of its arguments. Matrix multiplication is an example.

Let V, W and X be three vector spaces over the same base fieldF. A bilinear map is a function

$$B : V \times W \rightarrow X$$

such that for any w in W the map

$$v \mapsto B(v, w)$$

is a linear map from V to X, and for any v in V the map

$$w \mapsto B(v, w)$$

is a linear map from W to X.

In other words, if we hold the first entry of the bilinear map fixed, while letting the second entry vary, the result is a linear operator, and similarly if we hold the second entry fixed. Note that if we regard the product V × W as a vector space, then B is not a linear transformation of vector spaces (unless V = 0 or W = 0) because, for example $B(2(v,w)) = B(2v,2w) = 2B(v,2w) = 4B(v,w)$.

If V = W and we have $B(v,w) = B(w,v)$ for all v, w in V, then we say that B is symmetric.

The case where X is F, and we have a bilinear form, is particularly useful (see for example scalar product, inner product and quadratic form).

The definition works without any changes if instead of vector spaces over a field F, we use modules over a commutative ringR. It also can be easily generalized to n-ary functions, where the proper term is multilinear.

For the case of a non-commutative base ring R and a right module M_R and a left module $_RN$, we can define a bilinear map $B : M \times N \rightarrow T$, where T is an abelian group, such that for any n in N, $m \mapsto B(m, n)$ is a group homomorphism, and for any m in M, $n \mapsto B(m, n)$ is a group homomorphism too, and which also satisfies

$$B(mt, n) = B(m, tn)$$

for all m in M, n in N and t in R.

Attribute based encryption is proceeded by bilinear mapping of attribute information of data owner and the data to be stored in the cloud. Bilinear mapping process achieved by multiplicative factors of both Logical

IJSER

AND and XOR operations.In this HASBE scheme, a data encryptor specifies an access structure for a ciphertext which is referred to as the ciphertext policy. Only users with decryption keys whose associated attributes, specified in their key structures, satisfy the access structure can decrypt the ciphertext.

It is the process of pairing up the attribute information and thus cipher text policy ABE is processed.

D: **Master and Secret key generation**

Master key is generated by doing the Logical AND operations of given attributes of data owner. Using the masterkey, public key is generated and secret key is generated by doing the logical XOR operations. Ciphering algorithms are applied using the secret key, thus secured secret key is generated by Attribute based encryption.

E: **Secured Cloud Storage**

The Data owner's files are provided sufficient security. These files are stored in the cloud servers. In order to do that the cloud server have to configure using VMware tool. In cloud servers client files are stored as secured files so the crypto process can be applied. For crypto process we use Blowfish algorithm for the encryption and decryption process.

F: **Secure data retrieval**

Data retrieval process not only consists of retrieval of encrypted files from the cloud server and decrypted using respected private keys. But the data are provided to the users upon the authentication of the hierarchical access control of Cloud system architecture. Data or keys are relocated in the cloud frequently depending upon the kind of data owner's identity and the data to be stored on the cloud.

G: **Blowfish Algorithm**

Blowfish is a keyed, symmetric block cipher, included in a large number of cipher suites and encryption products. Blowfish provides a good encryption rate in software and no effective cryptanalysis of it has been found to date. Blowfish is a general-purpose algorithm, intended as an alternative to the ageing DES and free of the problems and constraints associated with other algorithms. Blowfish is a fast block cipher, except when changing keys. Each new key requires pre-processing equivalent to encrypting about 4 kilobytes of text, which is very slow compared to other block ciphers. This prevents its use in certain applications, but is not a problem in others. In one application, it is actually a benefit: the password-hashing method used in OpenBSD uses an algorithm derived from Blowfish that makes use of the slow key schedule; the idea is that the extra computational effort required gives protection against dictionary attacks. Blowfish has a memory footprint of just over 4 kilobytes of RAM. In this system the Blowfish algorithm is used for the key generation for encryption.

IV. CONCLUSION

The HASBE scheme incorporates a hierarchical structure of system users by applying a delegation algorithm to ASBE. HASBE not only supports compound attributes due to flexible attribute set combinations, but also achieves efficient user revocation because of multiple value assignments. In this project, we address this open issue and propose a secure and scalable fine-grained data access control scheme for cloud computing. Our proposed scheme is partially based on our observation that, in practical application scenarios each data file can be associated with a set of attributes which are meaningful in the context of interest. The access structure of each user can thus be defined as a unique logical expression over these attributes to reflect the scope of data files that the user is allowed to access. As the logical expression can represent any desired data file set, fine-grained of data access control is achieved. To enforce these access structures, we define a public key component for each attribute. Data files are encrypted using public key components corresponding to their attributes. User secret keys are defined to reflect their access structures so that a user is able to decrypt a cipher text if and only if the data file attributes satisfy his access structure.

In future I would like to go for enhancing the scalability and flexibility according to the user accessibility, reduce the complexity of the existing algorithm. And also provide the accessibility of data to multiple users with single trusted authority.

REFERENCES

[1] Allison Lewko, University of Texas at Austin alewko" New Proof Methods for Attribute Based Encryption: Achieving Full Security througt Selective technique"

[2] John Bethencourt Carnegie Mellon University and Amit Sahai "Ciphertext-Policy Attribute Based Encryption"

[3] [Junbeom Hur and Dong Kun Noh,Member,IEE"Attribute- Based Access Control with Efficient Revocation in Data Outsourcing System"

[4] Mariana Raykova,Hang Zhao,and Stevev M.Bellovi Coiumbia University, USA, fmariana,zhao,"privacy Enhanced Access Control for Outsourced Data Sharing"

[5] Melissa Chase Computer Science Department Brown University providence, RI 02912 "Multi-Authority Attribute Based Encryption"

[6] Melissa Chase Microsoft Reserch ,1 Microsoft Way Redmond,WA 98052,USA and Sherman S.M Courant Institution of Mathematical Science New York University, "improving Privacy and Security in Multi-Authority Attribute-Based Encryption"

[7] Ming li Member, IEEE, Shucheng Yu, Member, IEEE, Yao Zheng, IEEE,Kui Ren , Member, IEEE, and Wenjing Lou, Member, IEEE "Scalable and Secure Sharing of Personal Helth records in Cloud Computing using Attribute-Based Encryption"

[8] Sascha M uller, Stenfan Katzenbeisser, and Claudia Eckert Technische

IJSER

University at Darmstadt Hochschulstr." Distributed Attribute-Based Encryption"

[9] Shucheng Yu,Member,IEEE,Kui Ren, Member,IEEE,and Wenjing Lou, "FDAC: Toward Fine-Grained Distributed Data Access Control in Wirless Sensor Networks"

[10] Shucheng Yu, Cong Wang, Kui Renand Wenjing Lou,Dept. of ECE, Worcester Polytechnic Institute, Dept. of ECE, Illionis Institute of Technology. " Achieving Secure, Scalable, and Fine –grained data Access Control in Cloud Computing"

IJSER