A Secured File Store in Cloud Environment Using Hierarchy Attribute-Based Encryption

M.Kiruthika[1], R.Mohanabharathi[2]

[1]PG Scholar, Department of Computer Science and Engineering, Selvam College Of Technology, Namakkal, India
[2]Assistant Professor, Department of Computer Science and Engineering, Selvam College Of Technology, Namakkal, India

Abstract**—***Cloud Computing(CC) has been envisioned as the next production architecture of Information Technology (IT)Enterprise. In contrast to accepted solutions, anywhere the IT services are under proper physical, logical and personnel controls.CC moves the application software and databases to the max data centers, where the organization of the data and services may not be fully dependable. With CC and storage services, data is not only stored in the cloud, but routinely shared among a max number of users in a group. In this project, Hierarchy Attribute-Based Encryption(HABE) scheme is proposed for shared data with large groups in the cloud. Hash signatures are used to compute verification information on shared data, so that the authority is able to audit the correctness of shared data, but cannot reveal the identity of the signer on each block. Hash signature and Keys are generated by hierarchical access tree .Implementation of auditing scheme to perform efficient public to protect both identity and data privacy in cloud environments. Also users can access the data from data owner through cloud provider in real time dynamic cloud environment.*

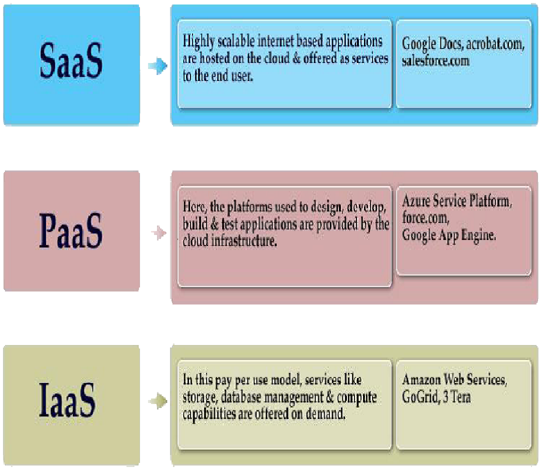
Keywords**—*****cloud data, cloud authority, security, cloud storage, auditing strategy.***

I. INTRODUCTION

CC is an emerging technology which provider a lot of opportunities for online sharing of resources or services. One of the fundamental advantages of CC is pay-as you-go pricing model, where customers pay only according to their usage of the services. Cloud Computing is an internet oriented computing.

Fig.1: CC Architecture

It dynamically delivers all as a service over the internet base and on user demand, such as network, storage, operating system, hardware, software and resources. These are cloud services types:



SaaS	Highly scalable internet based applications are hosted on the cloud & offered as services to the end user.	Google Docs, acrobat.com, salesforce.com
PaaS	Here, the platforms used to design, develop, build & test applications are provided by the cloud infrastructure.	Azure Service Platform, force.com, Google App Engine.
IaaS	In this pay per use model, services like storage, database management & compute capabilities are offered on demand.	Amazon Web Services, GoGrid, 3 Tera

Fig.2: CC Services

1.1 Software as a Service In this model, a full application is to be had to the customer, as a service on demand. Alone instance of the service runs on the cloud & multiple end users are serviced. On the patron's side, there is no need for open investment in servers or software license, while for the provider, the costs are lower, since only a single request needs to be hosted & maintained. Today SaaS is offered by company such as Google, Salesforce, Microsoft, Zoho, etc.

1.2 Platform as a Service Here, a layer of software or development environment is encapsulated & accessible as a service, upon which other top level of service can be built. The purchaser has the freedom to build his own application, which process on the provider's infrastructure. To meet manageability and scalability supplies of the applications, PaaS provider present a predefined group of OS and application servers, such as LAMP platform (Linux, Apache, MySql and PHP), Ruby etc. Google's App Engine, Force.com, etc are some of the popular PaaS examples.

1.3 Infrastructure as a Service IaaS provides basic resources and computing capability as standardized services over the network. Servers, storage space systems, networking equipment, data centre space etc. are shared and made available to handle workloads. The client would classically install his own software on the infrastructure. Some general examples are Amazon, GoGrid, 3 Tera, etc.

CC is application as three types such as Public, Private and Hybrid Clouds.

Public Cloud

Public clouds are owned and operate by third party; they distribute superior economies of range to patrons, as the infrastructure costs are reach among a mix of users, giving each character client an beautiful low-cost, "Pay-as-you-go" model. All clients share the same infrastructure pool with limited pattern, security protections, and ease of use variances. These are managed and support by the cloud provider. One of the compensation of a Public cloud is that they may be better than an enterprises cloud, thus providing the ability to scale effortlessly, on demand.

Private Cloud

Private clouds are built completely for a single project. They aim to address concern on data security and offer larger control, which is naturally lacking in a public cloud. There are two variations to a private cloud using follow:

- ✓ On-premise Private Cloud
- ✓ Externally hosted Private Cloud

On-premise Private Cloud.

On-premise private clouds, also known as familial clouds are hosted surrounded by one's own data center. This model provides a more even method and protection, but is limited in aspect of size and scalability. IT department would also need to incur the capital and prepared costs for the corporal resources. This is best right for applications which need complete control and configurability of the infrastructure and defense.

Externally hosted Private Cloud

This type of personal cloud is hosted externally with a cloud provider, where the provider facilitates an exclusive cloud setting with full agreement of privacy. This is best suited for enterprise that doesn't prefer a public cloud due to giving out of physical resources.

Hybrid Cloud

Hybrid Clouds connect equally public and private cloud model. With a Hybrid Cloud, examine providers can utilize 3rd party Cloud Providers in a full or partial manner thus increasing the suppleness of computing. The Hybrid cloud environment is capable of providing on-demand, on the exterior provisioned scale. The ability to enlarge a private cloud with the wealth of a public cloud can be used to administer any unexpected surges in workload.

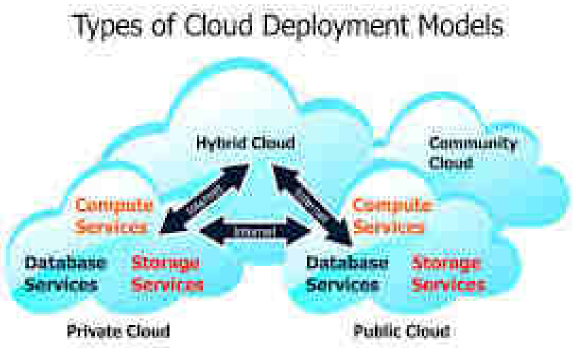


Fig.3: CC deployments

Cloud Storage system, is also recognized as Data storage as a service (DAAS), is the abstract of resources last an interface where resources can be administered on order. Cloud data resources works on distribution file systems because of its ability to handle a countless volume of data effectively. Storage can be limited or remote. Cloud computing is cost effective, safe and scalable but organization the load of random job available is a tricky work. Data ease of use means data is available when never it is request. Accessibility of data increases with augmentation in number of duplication of data. But after accomplishment a specific level of repetition, there occurs no growth in availability. So it is improved to find an optimum level of duplication. Availability and replication ratio also depends on node malfunction ratio. If failure probability is towering, more number of replication of that data is required. So if node breakdown ratio is less, less duplication number is necessary for maximum file availability.

CC BENEFITS

Enterprises would want to align their applications, so as to develop the architecture model that CC offers. Some of the classic benefits are listed follows:

- ✓ Reduced Cost
- ✓ Increased Storage
- ✓ Flexibility

Reduced Cost

There are a number of reasons to attribute Cloud technology with lesser costs. The billing model is pay as per usage; the infrastructure is not purchase thus lower maintenance. Initial cost and recurring fixed cost are much inferior than traditional computing.

Increased Storage

With the huge Infrastructure that is offered by Cloud provider today, storage & continuance of large volumes of data is a reality. Impulsive workload spike are also managed successfully & efficiently, since the cloud can scale dynamically.

Flexibility

This is an enormously important quality. With enterprises have to adapt, even more rapidly, to varying business conditions, speed to deliver is significant. Cloud computing stresses on getting application to market very rapidly, by using the majority appropriate building blocks compulsory for deployment.

II. LITERATURE SURVEY

2.1 Bethencourt J et al.[1] complex access control on encrypted data that is called as Cipher text-Policy Attribute-Based Encryption. By using these techniques encrypted data can be kept congenital even if the storage server is unfrosted; moreover, our methods are privacy against complexity attacks. Previous Attribute-Based Encryption systems used attributes to describe the encrypted data and built policy into user's keys while in our system attributes are used to describe a user's commendation and a revelry encrypting data determines a rule for who can decrypt. Thus, our methods are theoretically closer to traditional access manage methods such as Role-Based Access Control (RBAC). In addition, we provide an implementation of our system and give performance capacity

2.2 Goyal V et al.[2] In an ABE system, a user's keys and cipher texts are label with sets of descriptive attribute and a fastidious key can decrypt a particular cipher text only if there is a competition between the attribute of the cipher text and the user's key. The cryptosystem of Shay and Waters allowed for decryption when at least *k* attributes overlap between a cipher text and a private key. While this primitive was shown to be useful for error liberal encryption with biometrics, the lack of impressibility seems to limit its applicability to larger systems. In our cryptosystem, cipher texts are labeled with sets of attribute and private keys are connected with contact structure that manage which cipher texts a user is intelligent to decrypt. We disclose the applicability of our structure to sharing of audit log data and broadcast encryption. Our structure supports delegation of private keys which subsume Hierarchical Identity-Based Encryption (HIBE).

2.3 Joseph K. Liu et al.[4] In a fine-grained two-factor access control protocol for web-based cloud compute services, using a trivial security device. The device has the following properties:

(1) It can compute some light algorithms, e.g. hash and exponentiation; and

(2) It is tamper resistant, i.e., it is assumed that no one can smash into it to get the secret information stored inside.

With this device, our protocol provides 2FA privacy. First the user secret key (which is more often than not stored inside the computer) is required. In addition, the security device should be also coupled to the computer (e.g. through USB) in order to authenticate the user for access the cloud. The user can be granted access only if he has both items. Furthermore, the user cannot use his secret key with another device belong to others for the access. Our protocol chains fine-grained attribute-based access which provides a great suppleness for the system to set varied access policy according to different scenarios. At the same time, the privacy of the user is also preserved. The cloud system only knows that the user possesses some required attribute, but not the real identity of the user.

2.4 Tsz Hon Yuen et al.[7] ABE only deals with authenticated access on encrypted data in cloud storage service. It is not practical to deployed in the case of access control to cloud computing service: The cloud server may encrypt a chance message using the access policy and asks the user to decrypt it. If the user can successfully decrypt the cipher text, it is allowed to access the cloud computing service. Although this approach can fulfill the requirement, it is highly inefficient. In this new idea, a user can validate him/herself to the cloud calculate server incognito. The server only knows the user acquire some necessary attribute, yet it does not be common with the identity of this user. In supply a k-times edge for anonymous access control. That is, the member of staff serving at table may limit a meticulous set of user (i.e., person's users with the same set of quality) to access the system for a maximum k-times within a period or an event. Further other access will be denied. We also prove the privacy of our instantiation. Our success result shows that our scheme is practical.

2.5 Liang K et al.[3] To achieve more flexibility on re-encryption, many variants have been proposed Proxy Re-Encryption(PRE), Identity-Based PRE (IBPRE), and Attribute-Based PRE (ABPRE). CPRE allows an encryption connected with a condition to be converted to a new cipher text tagged with a new form. The technologies of IBPRE and ABPRE are rather similar, and a main difference between them is ABPRE enjoys more expressiveness in data sharing. Furthermore, the above encryption is allowed to be misshapen to another cipher text associated with a new string by a semi trusted proxy to whom a re-encryption key is agreed. Nonetheless, the proxy cannot gain access to the underlying plaintext. This new ancient can increase the flexibility of users to delegate their decryption rights to others. We also prove it as fully chosen cipher text secure in the standard model.

III. SYSTEM ARCHITECTURE

CC service provider requires a system which can handle a large number of requests at a time. For processing the huge cloud of requests for data access permission, services need to be very available. System keeps many copies of the blocks of data on different nodes by duplicate. A large number of replication strategies for management of replicas have been implemented in traditional system. As a result of replication, data replications are stored on different data nodes for high consistency and ease of use. Replication factor for each data block and replica placement sites need to be determined at first. In existing support data can be lost so in this paper propose improved secure perform in ABE to protect the data from loss. It present efficient constancy as a service model, where a group of data owners that compose service provider can verify whether the data cloud update the data or not and design user function table to change status of split files with different metrics and proposed in fig 1

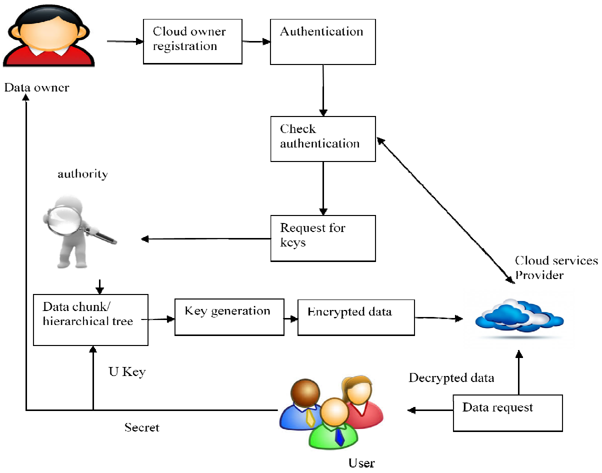


Fig.4: Improved Framework

IV. INDENTATIONS AND EQUATIONS

4.1 Setup ()

The setup algorithm take as input a privacy parameter λ and a small universe description U= {1, 2, 3…, ℓ}. It first runs G (λ) to obtain (p,G,G_T ,e), where G and and G_T are cyclic groups of prime order . It then chooses g,u,v,d ∑ G,and α,a ∑ Z_p^* uniformly at random,for each attribute i ∑ U. It chooses a random value Si ∑ Z* p and a collision-resistant hash function H:G⟶ Z_p^*,the public parameters PK = (G, G_T ,e,g,u,v,d,g^a , e(g, g)$^\alpha$,$T_i = g^{si}$ ∀ i ,H). It outputs a master public key and master secret key MSK= α.

4.2 KeyGen ()

The key generation algorithm randomly picks t ∑ Z* p

$K = g^\alpha g^{at}$

$K_o = g^t$

$K_i = T_i^t$ ∀ i∈ s

It outputs a transformation key and decryption key.

4.3 Encrypt ()

The encrypt algorithm use the public parameters, message and access structure. Access structure consists of attributes and their mapping.

$$C = u^{H(M)} v^{H(M)} d$$
$$C_1 = M.e(g.g)^{\alpha s}$$
$$C_1' = g^s$$
$$C_{1,i} = g^{a,A_i,v} T^{-r1,i} \rho(i)$$
$$D_{1,i} = g^{r1,i} \forall i \in \{1,2,\dots,1\}$$
$$C_2 = M.e(g.g)^{\alpha s}$$
$$C_2' = g^s$$
$$C_{2,i} = g^{a,A_i,v} T^{-r2,i} \rho(i)$$
$$D_{2,i} = g^{r2,i} \forall i \in \{1,2,\dots,1\}$$

It output a cipher texts CT as,

Encrypted data CT=((A, ρ),$\hat{c}$, $C_1, C_1', C_{1,i}, D_{1,i}, C_2, C_2', C_{2,i}, D_{2,i}$)

4.4 Transform ()

This algorithm will generate the transformed cipher text. This algorithm takes as input the public parameters PK, cipher text CT, and the transformation key TKs to generate the transformed cipher text CT’. It send the transformed cipher text to the user.

$$T_1' = [e(c_1', \frac{K'}{[(\prod_{i\in I}(e(C_{1,i}, K_0').e(K_{\rho(i)}', D_{1,i}))^{\omega i})}]$$

$$= [e(g,g)^{\alpha s/z} e(g,g)^{ats/z} / [\prod_{i\in I} (e(g,g)^{atAi.v\omega i/z}$$

$$= e(g,g)^{\alpha s/z}$$

$$T_2' = [e(c_2', \frac{K'}{[(\prod_{i\in I}(e(C_{2,i}, K_0').e(K_{\rho(i)}', D_{2,i}))^{\omega i})}]$$

$$= [e(g,g)^{\alpha s'/z} e(g,g)^{ats'/z} / [\prod_{i\in I} (e(g,g)^{atAi.v'\omega i/z}$$

$$= e(g,g)^{\alpha s'/z}$$

4.5 Decrypt ()

Decrypt algorithm uses the public parameters, transformed cipher text, and decryption key. PK = (G,G_T ,e,g,u,v,d,g^a ,e(g, g)$^\alpha$,$T_i = g^{si}$ =g ∀ i ,H)

CT=((A,ρ),$\hat{c}$, $\hat{c}$, $C_1, C_1', C_{1,i}, D_{1,i}, C_2, C_2', C_{2,i}, D_{2,i}$,i)

CT’ = (T=C, $T_1 = C_1, T_1', T_2' = C_2, T_2'$) .

RKs = z

4.6 Hierarchical Tree Algorithm

We present an implementation of the hierarchical tree algorithm on the individual time step algorithm (the Hermits scheme) for collision N-body simulations, running on GRAPE-9 system, a special future hardware accelerator for gravitational various body simulations. Such combination of the tree algorithm and the individual time step algorithm was not simple on the prior GRAPE system mainly as its recollection addressing scheme was limited only to sequential access to a full set of unit data. The present GRAPE-9 system has an indirect memory address unit and a particle memory big adequate to store all particle data and also tree nodes data. The indirect memory address unit stores dealings lists for the tree algorithm, which is construct on host computer, and, according to the contact lists, force pipelines calculate only the connections necessary. In our implementation, the interaction calculations are appreciably reduced compared to direct N^2 summation in the original Hermit scheme. For example, we can archive about a factor 30 of speedup (equivalent to about 17 teraflops) against the Hermits scheme for a simulation of N=10^6 system, using hardware of a peak speed of 0.6 teraflops for the Hermits scheme.

V. CONCLUSION & FUTURE WORK

In this paper, we currently enable data reliability proof and constancy services over multi cloud system using ABE which helps in informative violation as much as possible. The cloud reliability model and local auditing, global auditing that helps user to confirm the cloud service provider (CSP) provide the promised constancy or not and count the severity of the violations. Therefore system monitor consistency service model as well as level of data uploads which helps the user to get the data in updated version. User can recognize various sub servers in CSP. It is a considered to provide regular update mechanism to confirm fragments simply and provide the data to users after updation only.

REFERENCES

[1] J.Bethencourt, A.Sahai, and B.Waters, “Ciphertext-policy attributebased encryption,” IEEE Symposium on Security and Privacy, pp. 321– 334, May 2007.

[2] V.Goyal, O.Pandey, A.Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” Proceedings of the 13th ACM conference on Computer and communications security, pp. 89–98, October 2006.

[3] K. Liang, M. H. Au, J. K. Liu, W. Susilo, D. S. Wong, G. Yang, T. V. X. Phuong, and Q. Xie, “A DFA-based functional proxy re-encryption scheme for secure public cloud data sharing,” IEEE Transactions on Information Forensics and Security, vol. 9, no. 10, pp. 1667–1680, October 2014.

[4] J. K. Liu, M. H. Au, X. Huang, R. Lu, and J. Li, “Fine-grained twofactor access control for web-based cloud computing services,” IEEE Transactions on Information Forensics and Security, vol. 11, no. 3, pp. 484–497, March 2016.

[5] A. Sahai and B. Waters, “Fuzzy identity-based encryption,” Advances in Cryptology–EUROCRYPT, pp. 457–473, May 2005.

[6] X. Xie, H. Ma, J. Li, and X. Chen, “An efficient ciphertext-policy attribute-based access control towards revocation in cloud computing,”Journal of Universal Computer Science, vol. 19, no. 16, pp. 2349–2367,October 2013.

[7] T. H. Yuen, J. K. Liu, M. H. Au, X. Huang, W. Susilo, and J. Zhou, “ktimes attribute-based anonymous access control for cloud computing,” IEEE Transactions on Computers, vol. 64, no. 9, pp. 2595–2608, September 2015.