HABE: A SECURE FILE STORE IN CLOUD ENVIRONMENT

S.K.Dhosarathi[#1], V.Kiruthika[*2], B.Karthika[#3]

[#] Department of Computer Science and Engineering , K.Ramakrishnan College of Technology, Anna University, Trichy
[1]dhosarathi2198@gmail.com
[2]kiruthika842@gmail.com
[3]karthikanithya20@gmail.com

Abstract— **Cloud computing is an emerging technology which provider an assortment of opportunities for online sharing of resources or services. The large amount effective benefit of using cloud computing is high availability of services with less cost and simple scalability. Cloud provides different type of services and storage security but some challenges are still present in it. Out of which privacy concern, synchronization, scalability, load balancing and replication are important issues. Data replication means maintain many copies of similar data on similar server or on different servers. In connection with cloud computing data replication can be said as storing multiple copies of same data on different location (servers), locally or at remote sites. If data is present at one site only, then it will be very difficult to handle the requests for accessing the data. Server will face a heavy load situation and system performance may degrade. So in this project Implements Heuristic Attribute Based Encryption (HABE) approach to fragment data sets which are uploaded by data owner and implement graph based approach to calculate the distances using T-Coloring method to predict the data nodes for placing fragmented data. This proposed approach is very useful to data owner for protecting data from attackers. Then we extend our approach for checking consistency in cloud system at the time file updating. And propose a heuristic auditing strategy (HAS) which adds appropriate reads to reveal as many violations as possible. It can be done by using user operation table. Each user maintains a UOT for recording local operations. Each evidence in the UOT is described by three elements: operation, logical vector, and physical vector. Experimental results provide improved security and reduced retrieval time for accessing data from cloud system and implemented in real time cloud environments.**

Keywords— **Cloud Data, Data Fragmentation, Node placement, User operation, auditing strategy**

I. Introduction

Cloud computing 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 a computing. It dynamically delivers everything as a service over the internet base an on user demand, such as network, operating system, storage, hardware, software and resources. These are cloud services types: Infrastructure as a service (Iaas), Platform as a service (Paas), and Software as a Service (Saas). Cloud Computing is implementation as three models such as Public, Private and Hybrid Clouds. Cloud Storage system, is a also known as Daas (Data storage as a service), is the abstract of storage last an interface where resources can be administered on demand [7]. Cloud data resources works on sharing file systems because of its ability to handle an infinite volume of data effectively. Storage can be local or remote. Cloud computing is cost effective, secure and scalable but managing the load of random job available is a difficult work. Data availability means data is accessible when never it is requested. Accessibility of data increases with increment in number of duplication of data. But after reaching a specific level of duplication, there occurs no development in availability. So it is better to find an optimum level of duplication. Availability and duplication ratio also depends on node failure ratio. If failure probability is high, more number of duplication of that data is required. So if node failure ratio is less, less duplication number is required for maximum file availability.

A. Related work:

K. Bilal,et.al…,[1]presented a comparison of the major DCN architectures that address the issue of network scalability and oversubscription and simulated the performance of the major DCN architectures in various practicalscenario under different network configurations. The presentation of the three-tier architecture is dependent on physical topology and oversubscription share at different network layers.

M. Manzano,et.al…,[2]represents the robustness analysis of the DCN topologies and various observations. The consequences provide the standard heftiness metrics, such as average nodal degree, algebraic connectivity, and spectral radiuses provide the standard heftiness metrics, such as average nodal degree, algebraic connectivity, and spectral radiuses are incapable to evaluate DCNs appropriately. *Most* of the metrics only regarded as the largest connected component for robustness evaluation.

D. Boru,et.al…,[3]projected a data duplication technique for cloud computing with datacenters which save the energy, network bandwidth and communication delay both between geographically sharing data centers as well as inside each datacenter.

W. A. Jansen,et.al…, [9]formative the privacy of complex computer systems is also a ancient privacy problem that overshadows large scale computing in general. Attaining the high promise qualities in development has been an obscure goal of computer privacy researchers and practitioners, and is also a job in growth for cloud computing.

G. Kappes,et.al…,[10]analyze the privacy requirements in multitenant data systems. Then we introduce the dike authentication architecture, which combines native access control with renter namespace isolation that is backwards compatible to object-based file systems.

II. HABE FRAMEWORK:

Duplicate can be used for maintaining availability in company any load conditions or failure situations. By improving the technique of duplicate, performance and availability of system can be improved. But excessive duplicate can also adverse effects like high level storage cost or degradation in systems overall performance due to excessive use of bandwidth. So HABE framework is better to use because it can understand fragments of the data. T-coloring algorithm can provide improved results in case when system is in model state. It is generally used when requests are of similar nature and distributed equally. In T-coloring, measure the distances of each data for placing data in cloud system. Distances are calculated using centrality measure. Centrality is measure of the relative importance of a node in the network. But in HABE framework, data can be lost due to updating at the time of retrieving from cloud storage.This problem can be overcome by the following section and illustrated HABE framework in fig

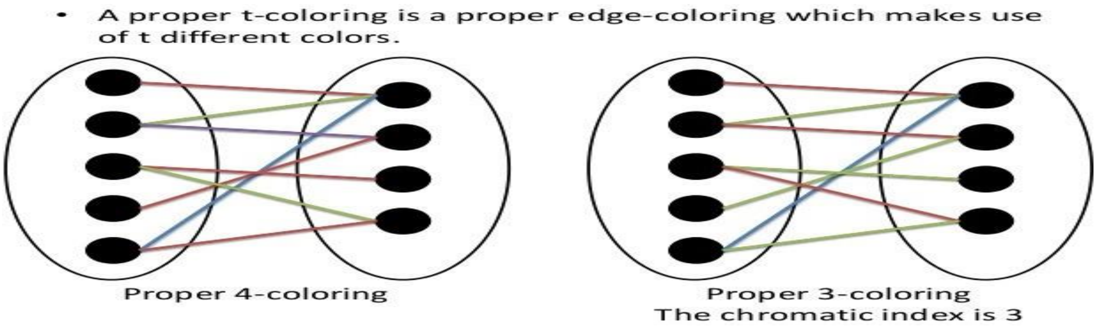
A proper t-coloring is a proper edge-coloring which makes use of t different colors.
Proper 4-coloring
Proper 3-coloring
The chromatic index is 3

Fig. 1 T-coloring Framework

III. TECHNIQUES FOR ABE ALGORITHM:

.

A. *Setup ()*

The setup algorithm takes as input a security 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.

B. *KeyGen ()*

The key generation algorithm randomly picks t ∑ Z* p

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

$K_o = g^t$

$K_i = T_i^t \forall\ i \in s$

It outputs a transformation key and decryption key.

C. *Encrypt ()*

The encrypt algorithm use the public parameters, message and access structure [17]. 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, $C_{2,i}$, $D_{2,i}$)

D. *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.

$$T'_1 = \left[e\left(c'_1, \frac{K'}{[(\prod_{i \in I} (e(C_{1,i}, K'_0).\, e(K'_{\rho(i)}, D_{1,i}))^{\omega i})}\right]$$
$$= [e(g,g)^{\alpha s/z} e(g,g)^{ats/z} / [\prod_{i \in I} (e(g,g)^{atAi.v\omega i/z}$$
$$= e(g,g)^{\alpha s/z}$$

$$T'_2 = \left[e(c'_2, \frac{K'}{[(\prod_{i\in I}(e(C_{2,i}, K'_0).e(K'_{\rho(i)}, D_{2,i}))^{\omega i})}\right]$$

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

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

E. Decrypt ()

Decrypt algorithm uses the public parameters, transformed cipher text, and decryption key.

$PK = (G, G_T, e, g, u, v, d, g^a, e(g, g)^{\alpha}, T_i = g^{si} = g \ \forall\ i, H)$

$CT = ((A, \rho), \hat{c}, \hat{c}, C_1, C'_1, C_{1,i}, D_{1,i}, C_2, C'_2, C_{2,i}, D_{2,i}, i)$

$CT' = (T = C, T_1 = C_1, T'_1, T'_2 = C_2, T_2')$.

RKs = z

IV. IMPROVED HABE FRAMEWORK:

Cloud computing service provider requires a system which can handle a large number of requests at a time. For processing the enormous cloud of needs for data access consent, 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 reliability and availability [16]. Duplication factor for each data block and replica placement sites need to be decided at first. In existing framework data can be lost so in this paper propose improved HABE framework that includes heuristic auditing strategy to protect the data from loss. It present efficient consistency as a service model, where a group of data owners that constitute service provider can verify whether the data cloud update the data or not and design user operation table to change status of fragmented files with different metrics and proposed framework in fig 2.

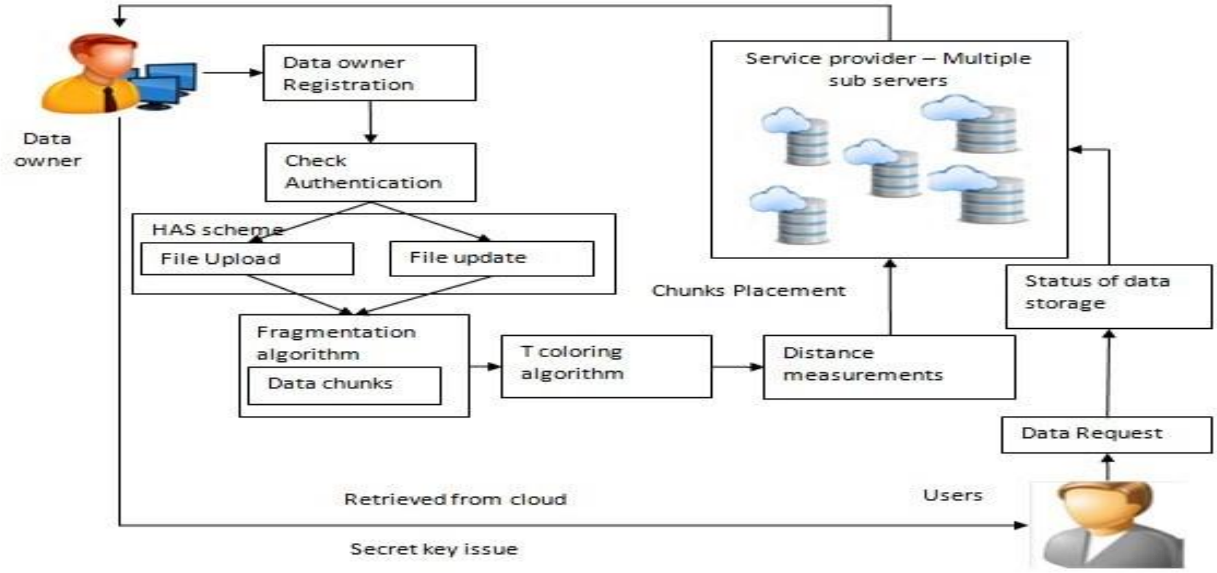

Fig. 2 Improved HABE Framework

V. Experimental Results:

We can evaluate the performance of the system using the parameters such as(i)increasing the number of nodes in the system, (ii)increasing the number of objects keeping numberof nodes constant, (iii) changing the nodes storagecapacity, and (iv) varying the read/write ratio. These measurements are consolidated as capacity of replication node and time of updation. And it can be plotted as graph in fig 3

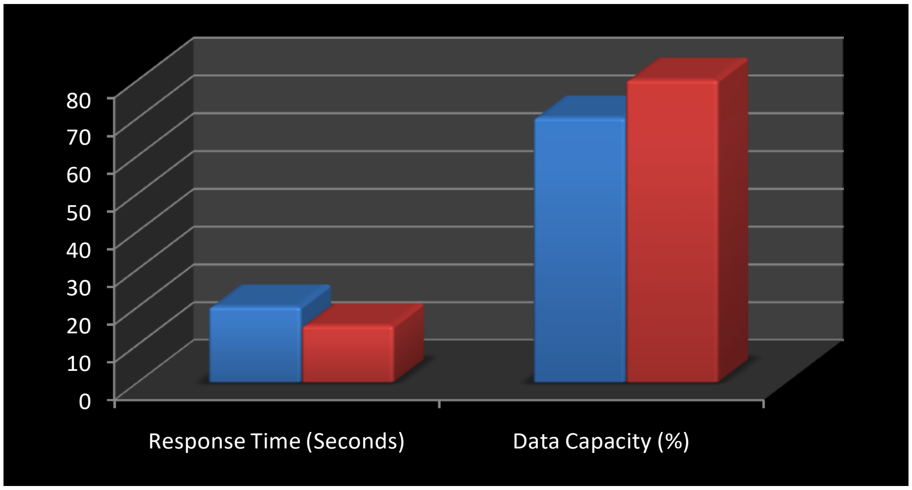

Fig. 3 Experimental Result

VI. Conclusion:

In this paper, we current enabling data reliability proof and regularity services over multi cloud system using Heuristic auditing strategy which helps in revealing violations as much as possible. The cloud consistency model and local auditing, global auditing that helps users to verify the cloud service provider (CSP) provides the promised consistency or not and quantify the severity of the violations. Therefore system monitors consistency service model as well as level of data upload which helps the user to get the data in updated version. User can understand various sub servers in CSP. It is a considered to provide regular update mechanism to authenticate fragments simply and provide the data to users after updation only.

References

[1] K. Bilal, S. U. Khan, L. Zhang, H. Li, K. Hayat, S. A. Madani, N. Min-Allah, L. Wang, D. Chen, M. Iqbal, C. Z. Xu, and A. Y. Zomaya, “Quantitative comparisons of the state of the art data center architectures,” Concurrency and Computation: Practice and Experience, Vol. 25, No. 12, 2013, pp. 1771-1783.

[2] K. Bilal, M. Manzano, S. U. Khan, E. Calle, K. Li, and A. Zomaya, “On the characterization of the structural robustness of data center networks,” IEEE Transactions on Cloud Computing, Vol. 1, No. 1, 2013, pp. 64-77.

[3] D. Boru, D. Kliazovich, F. Granelli, P. Bouvry, and A. Y. Zomaya, “Energy-efficient data replication in cloud computing datacenters,” In IEEE Globecom Workshops, 2013, pp. 446-451.

[4] Y. Deswarte, L. Blain, and J-C. Fabre, “Intrusion tolerance in distributed computing systems,” In Proceedings of IEEE Computer Society Symposium on Research in Security and Privacy, Oakland CA, pp. 110-121, 1991

[5] B. Grobauer, T.Walloschek, and E. Stocker, “Understanding cloud computing vulnerabilities,” IEEE Security and Privacy, Vol. 9, No. 2, 2011, pp. 50-57.

[6] W. K. Hale, “Frequency assignment: Theory and applications,” Proceedings of the IEEE, Vol. 68, No. 12, 1980, pp. 1497-1514.

[7] K. Hashizume, D. G. Rosado, E. Fernndez-Medina, and E. B. Fernandez, “An analysis of security issues for cloud computing,” Journal of Internet Services and Applications, Vol. 4, No. 1, 2013, pp. 1-13.

[8] M. Hogan, F. Liu, A.Sokol, and J. Tong, “NIST cloud computing standards roadmap,” NIST Special Publication, July 2011.

[9] W. A. Jansen, “Cloud hooks: Security and privacy issues in cloud computing,” In 44th Hawaii IEEE International Conference onSystem Sciences (HICSS), 2011, pp. 1-10.

[10] G. Kappes, A. Hatzieleftheriou, and S. V. Anastasiadis, “Dike: Virtualization-aware Access Control for Multitenant File systems,” University of Ioannina, Greece, Technical Report No. DCS2013-1, 2013.

[11] Y. Rouselakis and B. Waters, “Practical constructions and new proof methods for large universe attribute-based encryption,” in *Proc. ACMConf. Comput. Commun. Secur*., 2013, pp. 463–474.

[12] N. Attrapadung, J. Herranz, F. Laguillaumie, B. Libert, E. de Panafieu, and C. Ràfols, “Attribute-based encryption schemes with constant-size ciphertexts,” *Theoretical Comput. Sci*., vol. 422, pp. 15–38, Mar. 2012.

[13] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-policy attribute based encryption,” in *Proc. IEEE Symp. Secur. Privacy.* May 2007, pp. 321–334.

[14] B. Waters, “Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization,” in *Public Key Cryptography* (Lecture Notes in Computer Science), vol. 6571, D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, Eds. Berlin, Germany: Springer-Verlag, 2011, pp. 53–70.

[15] M. Green, S. Hohenberger, and B. Waters, “Outsourcing the decryption of ABE ciphertexts,” in *Proc. 20th USENIX Secur. Symp.,* 2011, p. 34.

[16] L. Cheung and C. Newport, “Provably secure ciphertext policy ABE,” in *Proc. ACM Conf. Comput. Commun. Secur.*, 2007, pp. 456–465.

[17] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” in *Proc. ACMConf. Comput. Commun.* Secur., 2006, pp. 89–98.

[18] A. Sahai and B. Waters, “Fuzzy identity-based encryption,” in *Advances in Cryptology*(Lecture Notes in Computer Science), vol. 3494,R. Cramer, Ed. Berlin, Germany: Springer-Verlag, 2005, pp. 457–473.

[19] J. Lai, R. H. Deng, C. Guan, and J. Weng, “Attribute-based encryption with verifiable outsourced decryption,” *IEEE Trans. Inf. ForensicsSecurity*, vol. 8, no. 8, pp. 1343–1354, Aug. 2013.

[20] S. Hohenberger and B. Waters, “Attribute-based encryption with fast decryption,” in*Public Key Cryptography*, 2013, pp. 162–179.