International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 06 | June 2020 www.irjet.net p-ISSN: 2395-0072

Achieving Efficient Data Deduplication and Key Aggregation Encryption System in Cloud

Dr.MK Jayanthi[1], P.Sri vaibhavi[2], P.V.Naga Saithya[3], Y.Harshitha Reddy[4]

[1]*Dr.MK Jayanthi Kannan, Professor, Department of CSE, FET, Jain University, Karnataka, India.*
[2]*Sri Vaibhavi.P ,B.Tech, Department of CSE, FET, Jain University, Karnataka, India.*
[3] *NagaSaithya.V.P, B.Tech, Department of CSE, FET, Jain University, Karnataka, India.*
[4]*Harshitha Reddy.Y, B.Tech, Department of CSE, FET, Jain University, Karnataka, India.*

Abstract - In cloud environments, users store their data or files in cloud storage but it's not infinitely large. In order to scale back the need of storage and bandwidth, data deduplication has been applied. Users can share one copy of the duplicate files or data blocks to eliminate redundant data. Besides, considering the privacy of sensitive files, the users hope that the cloud server cannot know any information about those files. They often use certain encryption algorithms to guard the sensitive files before storing them within the cloud storage. Unfortunately, previous schemes have a security problem.

1. INTRODUCTION

Cloud storage has become so popular during a now days not due to its cost efficient on demand usage but due to provision of giant amount of storage on which user can easily outsource its data. Most of the surveys told that almost half consumed storage on cloud is occupied by duplicate files or data. Challenge in front of today's cloud services is the management of that huge increasing quantity of data. To make data management scalable, de-duplication is introduced. Security and privacy are among top concerns for the general public cloud environments. This Data confidentiality for users is achieved with Convergent Encryption instead of Traditional encryption. With the help of Cloud the performance of storing data in computer will be increase day by day and access of data will be fast. When user stored the large amount of data in cloud it faces some challenges like data loss, infected application, data theft, privacy issue, data location, security at user level and data duplication. When data was in reading or writing mode and transfer from one network to another network on cloud then these issues damage the routine of system. So as to upgrade the consistency and accessibility and give disaster retrieval, information is by and large copied on various storage areas. The majority of this copied information applies an additional load on the capacity framework as far as extra space and transmission capacity to move the copied information on the system. Efficient information storage is serious and information deduplication system is considered as an empowering innovation for active storage of huge information. Deduplication method is a unique information compression strategy to remove the excess information.

2. SYSTEM MODEL

Our system is a two-level multi-domain deduplication model, which provides the intra-deduplication and inter-deduplication. More specifically, the first level contains a only number of domains D = fD1;D2; : : : ;Dng. Each domain Di corresponding to an organization (e.g., an enterprise or a university) contains a group of affiliated users (e.g., staff in an enterpriseor faculty and students in a university) and employs an agent to complete the intra-deduplication. The second level includes a cloud service provider, which conducts the inter-deduplication. Besides, a key distribution server is introduced to set up the system. Fig2.1 illustrates the system architecture of the proposed scheme and the details of each entity are described as follows.

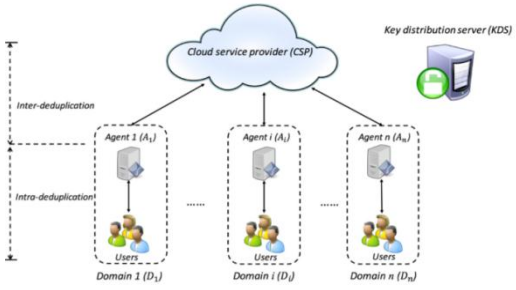

Fig2.1 system model

1. Key distribution server (KDS): The KDS is responsible for generating different private keys for users from different domains and a secret for the cloud service provider toperform the inter-deduplication.
2. Cloud service provider (CSP): The CSP offers storage services for users. Although the CSP seems to have abundant storage space, the corresponding costs of the management and maintenance for big data are relatively expensive. Hence, the CSP prefers to conduct the inter-deduplication over outsourced data from all domains to save costs by storing only one copy.
3. Users: Users from the same domain receive the same private key from the KDS, on the contrary, users from different domains possess different private keys. Based on the received private key, users can generate a random

convergent key used for encrypting the data and an intra-tag used for deduplicating.

4. Agent : In order to improve the efficiency of duplication search and resist an online brute-force attack launched by malicious users, an agent located between users and the CSP is hired by Di for performing intra-deduplication and transforming a random intra-tag into a random inter-tag.Theoverall process describes Where a user U from Di wants to upload data m, U generates a random intra-tag and sends it to Ai Then, Ai performs the intra-deduplication based on the received intra-tag. If a duplicate is found, Ai returns the corresponding feedback. Otherwise, agent transforms the intra-tag into a random inter-tag and sends it to the CSP for inter-deduplication. Note that only when the CSP does not find a duplicate, U needs to encrypt m and upload the corresponding ciphertext.

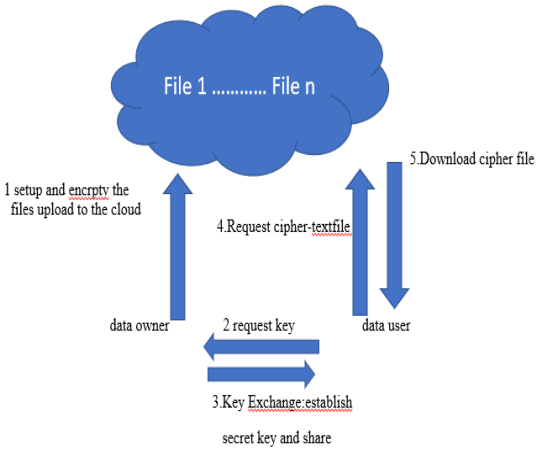

Fig2.2keyaggregation

3. B+ TREE

The B+ tree of order f is a very efficient, dynamic and balanced search tree that satisfies the following properties:

• The root node has at most f children and at least two children if it is not a leaf node.

• Each non-leaf node with nc children contains nc − 1 keywords: (P0, δ1, P1, δ2, P2, . . . , δnc−1, Pnc−1),where

– δk, k = 1, 2, . . . , nc−1, is a keyword with δk−1 <δk.

– Pk is a pointer that points to the root of the subtree. All the keywords pointed by Pk−1 are smaller than δk, but greater than or equal to δk−1.

– The number of children is satisfied [f/2]<=nc<=f.

• Each leaf node (unless it is a root) must contain nc − 1 keyword and pointer pairs: (δ1, P1, . . . , δnc−1, Pnc−1), where

– [f/2]-1<=nc-1<=f-1.

– All data is stored in leaf nodes, and the pointer Pk, k = 1, . . . ,nc − 1, points to the data that contains the keyword δk.

– Each leaf node has a link to its adjacent sibling, forming the ordered linked list.

• All leaves appear in the same level.

4. SYSTEM SETUP

The trusted KDS generates private keys for users, a secret for the CSP and the corresponding system parameters. Specifically, the KDS first runs the composite bilinear parameter generator Gen(κ0) to output a tuple (N = pq, G, GT ,e). Then, the KDS generates system parameters and private keys as follows.

• Randomly choose two generators g, x ∈ G and two numbers α, γ ∈ZN , and then compute y = x q , ν = g γ and gi = g α i for i = 1, 2, . . . , n, n + 2, . . . , 2n.

• For all users in Di , compute the private key di as di = g γ i , where i = 1, 2, . . . , n. Note that n is the total number of domains in system and i is the identifier of Di .

• Choose three cryptographic dddes : h1 : {0, 1} ∗ → {0, 1} κ1 , h2 : {0, 1} ∗ → {0, 1} κ0−1 and h3 : G → {0, 1} κ1 , where κ1 is the bit length of the convergent key. Finally, the KDS sends di to all users in Di and p to the CSP by secure channels, and publishes system parameters pp = (N, G, GT , e, h1, h2, h3, y, g, g1, . . . , gn, gn+2, . . . , g2n, ν).

In addition, based on the published system parameters, each agent Ai randomly chooses ai ∈ ZN as the private key, and then computes the corresponding public key g ai . Finally, Ai sends g ai to all users who belong to the domain Di. Note that in the practical scenario, it is important to support the dynamic addition of domains or users, especially for adding users, e.g., hiring new employees. The proposed scheme can easily support the addition of users to the existing domains. Specifically, suppose that a user U^* is added to the domain Di , we can introduce a simple identity verification protocol that allows U^* to verify

the legitimacy to the KDS. If the identity verification is passed, the KDS sends the corresponding private key di to U^* through the secure channel. After that, U^* can upload data in the same way as the users previously added to Di . For the addition of new domains, it seems impossible to increase n (i.e., the number of domains) after the scheme is once instantiated. However, similar to the concept of initializing arrays, we can initialize a relatively large n to reserve enough space to support the dynamic addition of domains to some extent. For example, if the number of domains is 10 in the current situation, then we can set n to 20. The reserved portion can be used for subsequent domain additions.

4.1 TAG GENARATION

When a user U from Di wants to upload data m, U chooses a random number rm ∈ ZN, and generates a random intra-tag τm with the private key d_i and A_i's public key g_{ai} as

$$\tau_m = \left(d_i^{h_2(m)} g^{a_i r_m}, g^{r_m}\right) = \left(g^{\alpha^i \cdot \gamma \cdot h_2(m) + a_i r_m}, g^{r_m}\right)$$

Then, U sends a message "upload k(L_m,τ_m)" to Ai. Note that L_m is the length of m, which acts as the search keyword of the deduplication decision tree (DDT) in the inter-deduplication.

INTRA-DEDUPLICATION

Upon receiving the message "upload k(L_m,τ_m)", Ai performs the intra-deduplication as follows.

1) Based on the private key ai, compute

$$\frac{d_i^{h_2(m)} g^{a_i r_m}}{\left(g^{r_m}\right)^{a_i}} = d_i^{h_2(m)} = g^{\alpha^i \cdot \gamma \cdot h_2(m)}.$$

Then, compute the dddes of

$$d_i^{h_2(m)} \text{ as } T_m = h_3\left(d_i^{h_2(m)}\right).$$

2) Check whether a duplicated copy of m exists by comparing Tm to previously stored dddes values from Di.

Algorithm 1 Search algorithm in the deduplication decision tree

Function: Search(L_m, node)

1: The search function Search(L_m, node) is started from the root node, i.e., node = root, and it proceeds to a leaf node as follows:
2: **if** node is a leaf **then**
3: **return** node
4: **else**
5: **case 1:** $L_m < \delta_1$ **then**
return Search(L_m, P_0)
case 2: $\delta_k \leqslant L_m < \delta_{k+1}$ **then**
return Search(L_m, P_k)
case 3: $\delta_{n_c-1} \leqslant L_m$ **then**
return Search(L_m, P_{n_c-1})
9: **end if**

INTER- DEDUPLICATION

After receiving the message "upload k(i,L_m, ^ τ_m)" from Di, the CSP performs the inter-deduplication to further eliminate the redundancy of data. Note that in the intra-deduplication, Ai judges whether the duplicate exists by comparing dddes values. Thus, the search complexity is very efficient, i.e., O(1). However, in the inter-deduplication, since inter-tags are random elements in G generated by BGN encryption, the same data will correspond to different inter-tags. Thus, the CSP cannot compare the dddes values of inter-tags to check the duplicate. Nevertheless, if the one by one search method is adopted, the time complexity of duplicate search increases linearly with the number of encrypted data stored in the CSP, which will lead to huge search costs, especially for big data. Accordingly, we construct a deduplication search tree (DDT) based on the B+ tree for efficiently searching duplicates.

Algorithm 2 Inter-deduplication algorithm

1: For the received data $(i, L_m, \hat{\tau}_m)$, the CSP calls the function Search(L_m, node) in the DDT to search whether the value L_m has already been stored.
2: **if** the same value is not found **then**
3: **return** "data upload"
4: **else**
5: the same value L_{m^*} is found, e.g., $(j, \hat{\tau}_{m^*}, B_{m^*}, C_{m^*})$, and then check whether $i = j$ holds.
6: **if** $i = j$ **then**
7: **return** "data upload"
8: **else**
9: verify whether the following equation holds

$$e(\hat{\tau}_m, g_j)^p = e(\hat{\tau}_{m^*}, g_i)^p \quad (4)$$

10: **if** Eq. (4) holds **then**
11: **return** "duplication$\|\text{link}_{m^*}\|B_{m^*}$"
12: **else**
13: **return** "data upload"
14: **end if**
15: **end if**
16: **end if**

KEY AGGREGATION

Key Aggregation:
FunctionKey aggregationGeneration(group name):
Input: Group name gn
Begins:
Read the group name as gn
Encrypt the gn using DDDES algorithm
Return E(gn) as key

FunctionKeyaggregationVerification(User Request,filegroup):
Input: Group name user request, filegroup
Begins:
Read the filegroup as gn
Encrypt the gni using DDDES algorithm
If userreq for a file then
If get the filegroup gnj of the user requested file then
Encrypt gnj using DDDES algorithm
If(userrquest group key and filegroup is same) then
Allow the user to download all the files in the group

4.1.1 Data Download

After a period of time, when the user U from Di wants to down-load an outsourced data m, U uses the

corresponding label labelm to find the stored Tm, and then sends a request "downloadkTm" to Ai. After receiving the request "download kTm", Ai finds the same dddes value and get the corresponding link linkm. Then,Ai returns linkm to U. After that, U can directly download the ciphertext Cm from the CSP based on the linkm.Finally, U decrypts Cm with the stored convergent key ckm and verifies data integrity. The details are introduced as follows.

• U recovers m0 by decrypting Cm with ckm.
• Then, with the recovered m0, U computes Tm0 =h3(dh2(m0)i), and verifies the integrity by checkingwhether Tm0 = Tm holds. If it does not hold, it meansthat m has been corrupted, i.e., m0 6= m.Note that in our scheme, if a user from Di has previouslytried to upload the data m. After the data deduplication, matterwhether m is successfully uploaded, Ai has a record of this data, i.e., (Tm, Bm, linkm). Conversely, if Ai does not store thecorresponding record, then it means that no user in Di has evertried to upload the data m, that is all users in Di do not have the access right to this data. Therefore, users only need to interactwithAito obtain the corresponding logical link, which can reducethe download time to a great extent compared to the interaction with the CSP.

4.1.2. MODULES

This project consists of the following modules:

1. Data Owner Registration
2. Data User Registration
3. Data user upload/download file

1. Data Owner Registration

• In this module, if a data owner has to view all the files uploaded by the users then he/she should login first.
• When the Owner is successfully logged in he/she can view all the files uploaded by the users and can check the status of the files.
• The owner can delete any file if he pleases to do so.

2. Data User Registration

• In this module, the user has to login to upload/download files.
• When the User is successfully logged in he/she can view all the files uploaded by him/her and can upload or download any files.

3. Data User Upload/Download file

• In this module, the user can view all his uploaded files and can download any of his file.
• If the user wants to delete a file then an OTP is sent to the user's mail id which should be correctly entered to delete the file.

4.1.3 DESIGN GOALS:

1. Confidentiality

Confidentiality refers to protecting information from being accessed by unauthorized parties .in other words only the people that are authorized to try to to so can gain access to sensitive data. Imagine your bank who are helping you with a transaction should be able to access them but no one else should. A failure to take care of Confidentiality means someone who shouldn't have access has managed to urge it through intentional behaviour or accidentally such a failure of Confidentiality,commonlyknow as a breach, typically cannot be remedied. Once the secret has been revealed, there's no way to un-reveal if. If your bank records are posted on a public website, everyone can know your checking account number,balance etc. And that information can't be erased from their minds,papers, computer and other places. Nearly all the main security incidents reported within the media today involve major losses of confidentiality.

2. Integrity

Integrity refers to making sure the authenticity of data that information isn't altered, which the source of the knowledge is genuine. Imagine that you simply have an internet site and you sell products thereon site. Now imagine that an attacker can shop on your website and maliciously alter the costs of your products , in order that they will buy anything for whatever price they choose. That would be a failure of integrity, because your information in this case the price product has been alteed and you didn't authorize this alteration. another example of a failure of integrity is once you attempt to hook up with an internet site and a malicious attacker between you and therefore the website redirects your traffic to a different website. In tis case the site you are directed to is not genuine.

3. Availability

Availability means information is accessible by authorized users. If an attacker isn't ready to compromise the primary two elements of data security they'll attempt to execute attacks like denial of service that might bring down the server, making the website unavailable to legitimate users thanks to lack of Availability.

4. Brute force attack

A brute force attack may be a trial and error method employed by application programs to decode encrypted

data like passwords or encoding Standard DES keys, through exhaustive effort (using brute force) rather than employing intellectual strategies. Just as a criminal might forced an entry , or "crack" a secure by trying many possible combinations, a brute force attacking application proceeds through all possible combinations of legal characters in sequence.A hacker may use a brute force attack to obtain access to a website and account, then steal data, shut the site down, or execute another type of attack. Brute force is taken into account to be an infallible, although time-consuming, approach.

5. Cloud admin

After login if the user uploads a file it'll not only be stored in the user's account but also be stored with cloud admin, but cloud admin won't be able to open files because the user will be having both private and public key and the admin will be having only public key. The user's files will be uploaded to cloud only when admin approves our files and other users will also be able to view the files.

6. Cloud server

After the user registers and logs in the user won't be able to access any file. the user will be able to access all the files only when the cloud server activates our files, then the user will be having access to upload or download or even be able to update the files.

5. DATA FLOW

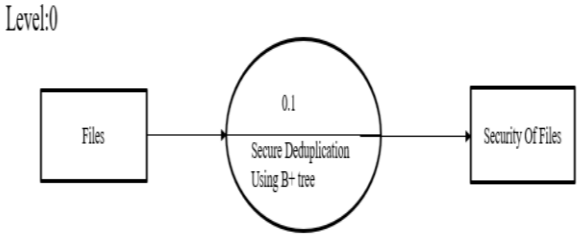

Fig4.5.1Process at level0

In level 0 it describes the whole process of dedupliaction and key aggregation process.

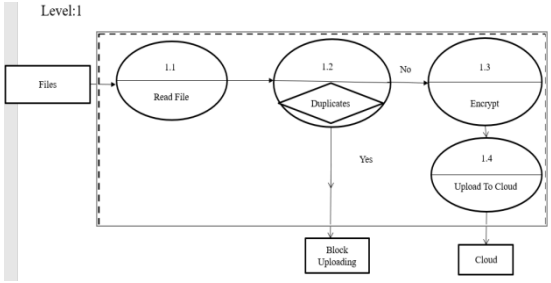

Fig4.5.2 Process at level1

In level1 it describes about deduplication process where the system checks the file whether it duplicate or not if it is duplicate it blocks uploading if not, it will encrypt and upload to the cloud.

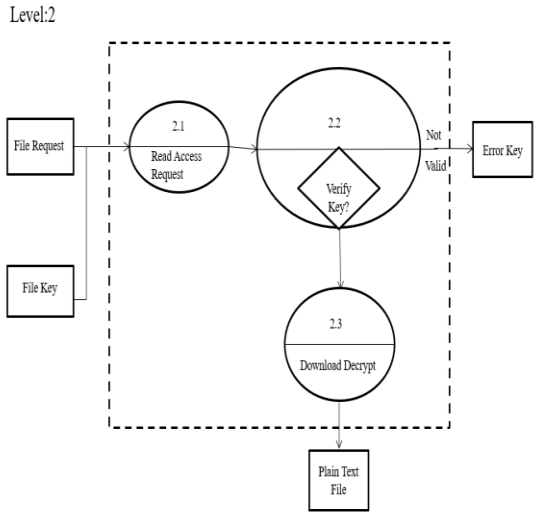

Fig4.5.3 Process at level2

In level 2 the data user request for the file and owner will send the key and it verify the key if it is correct then it will be downloaded or else it will show error.

6. CONCLUSION

Management of data has become very common for cloud services. Cloud services prefer to focus on their core business than to manage huge amount of data getting added each day. This work studied various aspects of deduplication of data. The various needs of the cloud services such as the data management, data deduplication, and encryption were studied. The project implemented a scenario where the cloud service can deduplicate the uploaded data. The data users are provided with the proper data. The data was prevented from unwanted exposure and unauthorized access by placing a proper access control mechanism. Authorized data deduplication aims at data security to keep the data secured and avoid unauthorized access. Deduplication at encryption level saves a lot of memory and memory can be utilized efficiently

7. REFERENCES

[1] D. Harnik, B. Pinkas, and A. Shulman-Peleg, “Side channels in cloud services:
Deduplication in cloud storage,” IEEE Security & Privacy, vol. 8, no. 6, pp. 40–47, 2010.
[2] D. T. Meyer and W. J. Bolosky, “A study of practical deduplication,” TOS, vol. 7, no. 4,
pp. 14:1–14:20, 2012.
[3] Pooja More, D G Harkut, “Cloud Data Security using Attribute-based
KeyAggregate Cryptosystem”,2016.

[4] S. Keelveedhi, M. Bellare, and T. Ristenpart, "Dupless: Server-aided
encryption for deduplicated storage," in Proceedings of the 22th USENIX Security
Symposium, Washington, DC, USA, August 14-16, 2013, 2013, pp. 179–194.
[5] Xue Yang, Rongxing Lu, Senior Member, IEEE, Jun Shao, Xiaohu Tang,
Member, IEEE and Ali A. Ghorbani, Senior Member, IEEE,"Achieving
Efficient and Privacy-Preserving Multi-Domain Big Data Deduplication in
Cloud",2018.