CONSTRUCTION OF A SECURE HYBRID CLOUD USING DE-DUPLICATED APPROACH

BELLALA ANUSHA [#1] , V. SRIKANTH [#2]

[#1] M.Tech Scholar ,Department of CSE
Pydah College of Engineering and Technology, Gambheeram Village,
Anandapuram, Mandal-531163. Visakhapatnam, AP, India.

[#2] Assistant Professor , Department of CSE
Pydah College of Engineering and Technology, Gambheeram Village,
Anandapuram, Mandal-531163.Visakhapatnam, AP, India.

ABSTRACT

Currently cloud computing has become one of the fascinating domains which were almost used by various IT companies. A cloud server is formed by connecting a various number of systems all together for a centralized remote server hosted on internet to store, modify and access the data to and from remote systems not from local machines. As the cloud has become one of the fascinating domain and attracted a lot of users towards that usage, but it still has some limitations in the current cloud service providers. First limitation is all the data which is stored on the cloud server is stored in the normal manner or in plain text so that it can be viewed and modified by anyone within the group. As we clearly understood that the data is going to be stored on remote server, whenever any user who want to access the data, he will retrieve the data from that remote server. In the current cloud servers there is no facility like avoiding the duplicate data to be stored into the cloud, so this lead a major problem like huge maintenance cost while storing the data inside the cloud server. So in order to avoid this knowledge/data duplication we'd like to use a replacement principle referred to as Data Deduplication.This is one among the simplest knowledge compression technique that was used for eliminating the duplicate copies of perennial knowledge and this was wide employed in recent cloud storage. So in this thesis we try to implement the live cloud service like **DRIVEHQ.com** with an enterprise cloud service account like **PYDAHCLOUD17** for storing the records in a encrypted manner inside that enterprise cloud account. As an extension for the current thesis we implemented a concept like monitoring of cloud activities like login and logout activities by admin as we know that admin is the role who can monitor all the user activities like upload, download and modify date and time details of the entire session, which is not available in any of the current cloud servers.

Key Words: De-Duplication, Encryption, CSP, Centralized Storage, Data Duplication, Authentication.

I. INTRODUCTION

In recent days cloud has occupied a major role in each and every part of the information processing and information storage centers. As the cloud has became a valuable resource for all parts of information processing centers, the data which is to be stored will be stored on the remote systems not on their local hardware, and accessed remotely via internet by connecting various servers. As the data will be stored on remote server, the data user need to retrieve the data from the remote server, whenever he want any data from that remote hardware. In the current cloud servers, the major limitation is data which is stored and shared over the cloud users has no security and there is also no security for accessing the data in the current cloud servers [1]. This is mainly because all the data which is stored in the current cloud servers is stored in the form of plain text rather than in a cipher text manner. As we know that cloud has exaggerated user attention in storing their valuable or sensitive information however limits in allocating resources dynamically. As we know that cloud has received more and more user's attention towards data storage, it still has some restrictions in size constraints. If we use any cloud in terms of Public account then we can afford a storage size of upto 1 GB only and if the same cloud we use in enterprise purpose we can use more that 1 TB and so on till unlimited data storage for a very large MNC companies which opt for storing their valuable data inside its specified location[2].

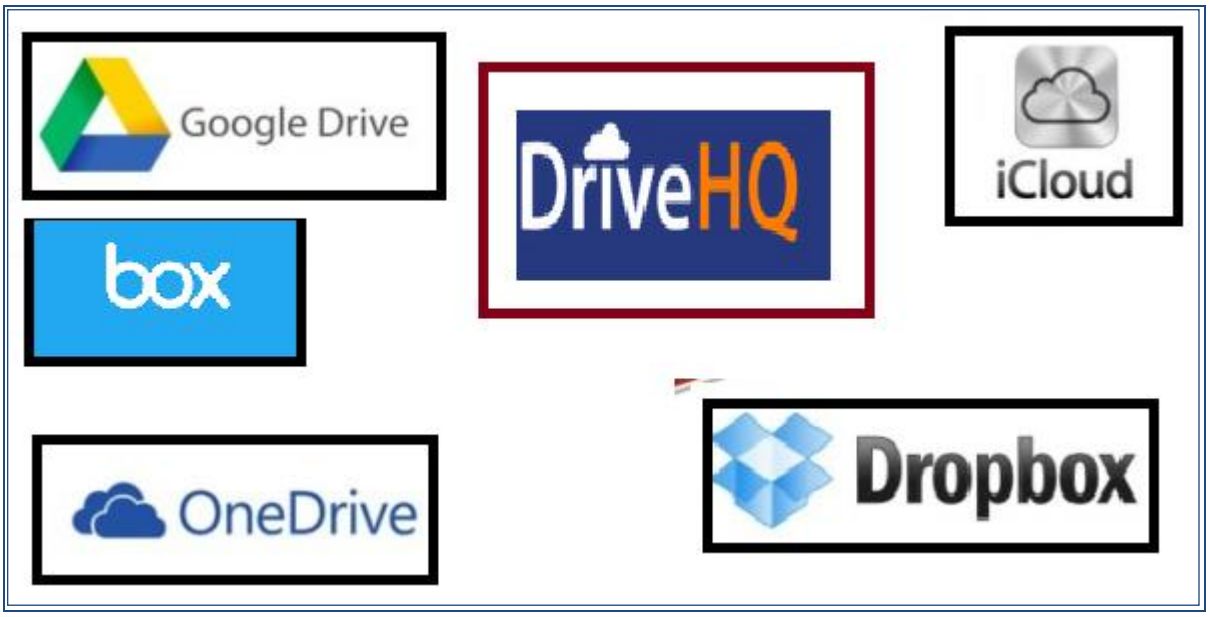


FIGURE.1. REPRESENTS THE VARIOUS CLOUD SERVICE PROVIDERS FOR STORING SENSITIVE INFORMATION

From the above figure 1, we can clearly find out that there are many cloud service providers that are available in the current days for storing and accessing the data remotely. Here in the above figure, there are some public clouds which takes no amount for storing the data till 1 GB of space ,but some clouds are their which will give access only for the premium members like those who have premium account. In this proposed thesis, we are going to use the DRIVEHQ.com as the cloud service provider for storing the sensitive information without any

duplicate data reside into its memory area. In this proposed thesis, we are using the DRIVEHQ.com the cloud service provider for storing and accessing the data in a secure manner with an enterprise public access account like "**PYDAHCLOUD17**" which can accept data up to 1 GB of space, freely and more than 1 GB it will be charging the end users from its side [3], [4].

Original plain data ought to be accessible only by reliable parties that do not embrace cloud suppliers, intermediaries, and Internet; in any untrusted context, data ought to be encrypted. Satisfying these goals has fully completely different levels of quality betting on the type of cloud service. There are several solutions guaranteeing confidentiality for the storage as a service paradigm, whereas guaranteeing confidentiality at intervals the data as a service (DaaS) paradigm [5] remains associate open analysis house. Throughout this context, we have a tendency to propose a brand new Secure DaaS as a result of the primary answer that allows cloud tenants to require full advantage of DaaS qualities, like handiness, dependability, and elastic quality, while not exposing unencrypted data to the cloud provider.

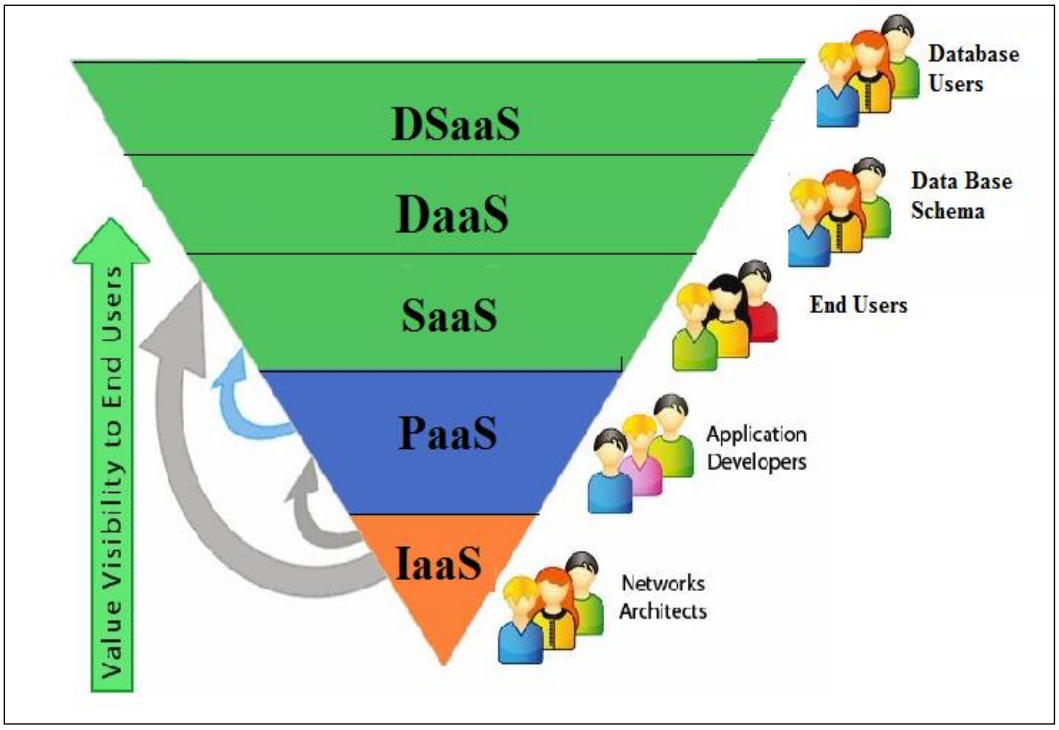


FIGURE. 2. REPRESENTS THE MANAGEMENT SYSTEM OF PRIMITIVE CLOUD SERVICES ALONG WITH DAAS AND DSAAS

From the above figure 2, we can clearly find out that management system of various primitive cloud services that are available in the literature. In the above figure we can clearly find out that there are various types of services that are available in the cloud computing. Our proposed thesis mainly comes under DaaS (I.e. Database as a Service) and in which the current data which is stored in cloud will be stored in plain text manner with no security .So as per the future scope of cloud services, they try to launch a new type of service like DSaaS (I.e. Database Security as a Service), where this has a facility to encrypt the data and then store the data into the database and in turn the file can be viewed by selected users only.

II. RELATED WORK

In this section we will find the related work that was analyzed and studied in order to implement this current thesis. This section will describe the work that is related to distinct data storage in the cloud database.

DISTINCT DATA STORAGE SERVICE

Distinct Data Storage Service (DDSS) is also termed as Single-instance storage service (SISS) may be a system's ability to stay one copy of content that multiple users or computers share. This is a process of eliminating the content that was duplicated and has the facility to extent the potency. It is basically enforced in the file systems, e-mail server code, knowledge backup and alternative storage-related solutions. In the case of associate degree e-mail server, single-instance storage would mean that one copy of a message is command inside its information while individual mailboxes access their own content with the help of a reference pointer. However, there's a typical idea that the first advantage of single instance storage in mail server solutions may be a reduction in space needs [6].

As we all know that main reason to use this DDSS is to greatly enhance the potency of messages sent to massive distribution lists. When employed in conjunction with a backup resolution, single instance storage will scale back the amount of archive media needed since it avoids storing duplicate copies of identical file. Typically identical files are put in on multiple computers, for instance OS files. With solutions that use single instance storage, only 1 copy of a file is written to the backup media thus reducing area. This becomes additional necessary once the storage is offsite and on cloud like Storage as a Service like Amazon S3.

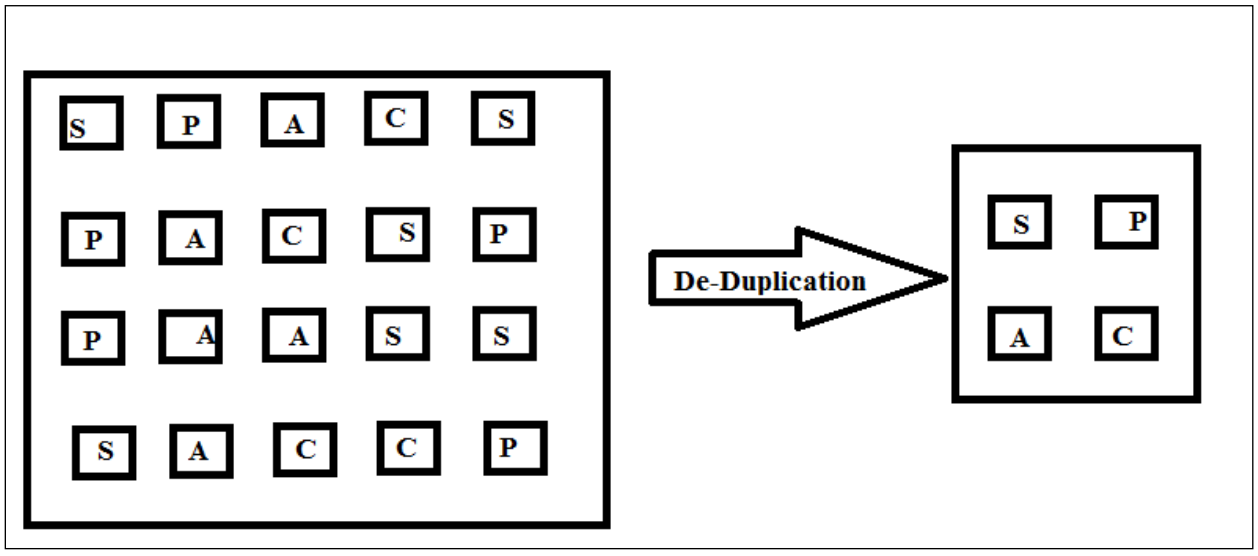


FIGURE. 3 REPRESENT THE SAMPLE ARCHITECTURE OF INFORMATION DE-DUPLICATION ON AN ALPAHBET PATTERN

From the above figure 3, we can clearly find out the architecture flow of a data de-duplication technique and its advantages over the data base servers. For instance we took various alphabets as input with S, P, A, C as 4 alphabets and initially we took alphabets in a duplicated manner. Once after applying the de-duplication or single instance service process all the duplicate letters are eliminated and only the distinct values like S, P, A, C are only retrieved as a result at the other end. This example clearly justifies how much the data de-duplication [6] has advantage over the various data base servers including the cloud database server [7].

III. A NOVEL PROTOCOL FOR SECURE DATA DE-DUPLICATION

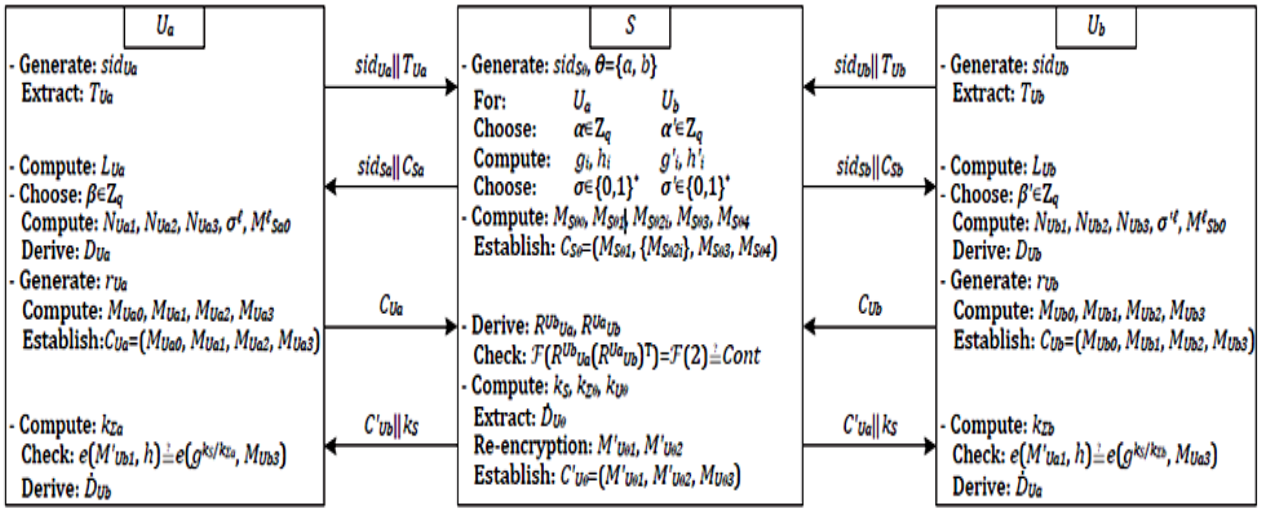
In this section, we mainly describe the proposed novel protocol for secure data de-duplication with all its facilities for storing the information on a public or private cloud in a secure manner as well as avoiding the duplicate copy of same data not to reside on the current cloud storage.

A) MAIN MOTIVATION

The cloud storage system includes a cloud server *S*, and users *{Ux}* (*x = {1….m}, m €* N*). There into, *Ua* and *Ub* are two users, which have independent access authorities on their own data fields. It means that a user has an access permission for particular data fields stored by *S*, and the user cannot exceed its authority access to obtain other users' data fields [8]. Here, we consider *S* and *{Ua, Ub}* to present the protocol phases for data access control and access authority sharing with enhanced privacy considerations [9]. The main notations are introduced in **TABLE 1.**

Notation	Description
S, U_x	The cloud server, and a user (i.e., cloud data owner).
PID_{U_x}	U_x's pseudorandom identifier (pseudonym).
T_{U_x}	U_x's identity token that is assigned by S.
sid_{S_x}, sid_{U_x}	The pseudorandom session identifier of S, U_x.
α, σ, β, r_{U_x}	The randomly generated numbers.
$R_{U_x}^{U_y}$	The access request pointer that represents U_x's access desire on U_y's data fields.
D_{U_x}, $\dot{D}_{U_x}$	U_x's own authorized data fields, and U_x's temp authorized data fields.
A_{U_x}, L_{U_x}, P_{U_x}	The data attribute access list, re-structure data access list, and data access policy.
$\{mpk/msk\}$	The pairwise master public/privacy keys.
$\{pk/sk\}$	The pairwise public/privacy keys.
k_{Σ_x}, k_{U_x}	The aggregated keys, and the re-encryption keys.
V^{ℓ}	The locally computed value V according to the same algorithm.
C_{S_x}, C_{U_x}	The ciphertexts.
$\mathcal{F}_{S_x}(x, P_{U_x})$	The defined polynomial owned by S.
$\mathcal{F}_{U_x}(x, L_{U_x})$	The defined polynomial owned by U_x.

The proposed algorithm has three major components like Ua ,S and Ub.Where Ua and Ub are two cloud users and S is cloud server. So now if we want to apply our proposed algorithm on this data. This is represented as follows:



A) Sender (Ua) and Receiver (Ub) Access Challenges and S's Access Response

Initially the source and destination {Ua, Ub} respectively generate the session identifiers {sidUa , sidUb}, extract the identity tokens {T_{Ua} , T_{Ub} }, and transmits {sidUa||TUa, sidUb ||TUa } to S as an access query to initiate a new session. Accordingly, we take the interactions of Ua and S as an example to introduce the following authentication phase. Upon receiving senders challenge, S first generates a session identifier for that source node sidSa , and establishes the master public key mpk = (gi , h ,hi ,BG , e(g , h) ,H) and master privacy key msk = (α ,g). There into, S randomly chooses α € Zq, and computes gi = $g\alpha^{i}$ and hi = $h\alpha^{i-1}$ (i = {1,…., n} € Z*).S randomly chooses σ €{0; 1}*, and extracts Ua's access authority policy P_{Ua} = $[pij]_{n\times m}$ (pij € {0; 1}), and Ua are assigned with the access authority on its own data fields DUa within PUa 's permission. S further defines a polynomial F_{Sa} (x , PUa) according to PUa and TUa [10]-[11].

$$\mathcal{F}_{S_a}(x, P_{U_a}) = \prod_{i=1,j=1}^{n,m} (x + ijH(T_{U_a}))^{P_{ij}} \pmod{q}$$

S computes a set of values {M_{Sa0}, M_{Sa1}, {M_{Sa2i}}, M_{Sa3}, M_{Sa4}} to establish the cipher text C_{Sa} = {M_{Sa1} , {M_{Sa2i}};M_{Sa3};M_{Sa4}}, and transmits sidSa || C_{Sa} to Ua.

Similarly, Ub performs the corresponding operations, including that Ub extracts A_{Ub} , and determines {L_{Ub} ,F_{Ub} (x , L_{Ub}), f_{Ubi}}. Ub further randomly chooses β′ € Zq, and computes the values {$N_{Ub}1$, $N_{Ub}2$, $N_{Ub}3$, σ′ℓ, $M\ell_{Ub}$ } to derive its own data fields D_{Ub} . Ub also extracts its meaning PID_{Ub} and an access request R_{Ua}^{Ub} to establish a cipher text C_{Ub} with the elements {$M_{Ub}0$;$M_{Ub}1$;$M_{Ub}2$;$M_{Ub}3$}.

B) { Ua, Ub}'s Access Request Matching and Data Access Authority Sharing

Upon receiving the cipher texts $\{C_{Ua}, C_{Ub}\}$ within an allowable time interval, and S extracts $\{PID_{Ua}, PID_{Ub}\}$ to derive the access requests $\{R_{ub}^{Ua}, R_{Ua}^{Ub}\}$.

$$R_{Ub}^{Ua} = H(sidSa \,||\, PIDUa) \pm M_{Ua}0;$$

$$R_{Ua}^{Ub} = H(sidSb \,||\, PIDUb) \pm MUb0:$$

S checks whether $\{R_{Ub}^{Ua}, R_{Ua}^{Ub}\}$ satisfy $F(R_{Ub}^{Ua} (R_{Ua}^{Ub})^{T}) = F(2) = Cont$.

If it holds, S will learn that both Ua and Ub have the access desires to access each other's authorized data [12], and to share its authorized data fields with each other. S extracts the keys $\{skS, pkUa, pkUb\}$ to establish the aggregated keys $\{kS, k\Sigma 0\}$ by the Diffie- Hellman key agreement [13], and computes the available re-encryption [14] key kU_{θ} for U_{θ} ($_{\theta \epsilon}$ $\{a; b\}$).

IV. ATTRIBUTE BASED ENCRYPTION(ABE) ALGORITHM

In this section we mainly discuss about the ABE algorithm that was used as a back end for granting access roles for the cloud data users for accessing the files by the private cloud. Here the file which was uploaded into the cloud server or public cloud server can be accessed only by the valid user who has authorized permission from the private cloud. If they don't have authorized permission to access that, then they will be treated as invalid and author or display a message likes "The user doesn't have the authorization Facility". Now let us discuss about the architecture of our proposed thesis and also find the explanation about the ABE in detail.

DESCRIPTION

The ABE algorithm is implemented as five polynomial-time algorithms, as follows. The Data owner establishes the public system parameter via Setup and generates a public/master-secret key pair via **Key Gen**. Messages are encrypted via Encrypt technique. The **Data Owner** can use the master-secret key to generate a decryption key for a set of cipher text classes via **Extract**. The generated keys are passed securely (via secure channels) and finally a user with an appropriate key can decrypt any ciphertext provided that the cipher text's class is contained in the storage node key via Decrypt method[15].

1. Here Data Owner is nothing but who will upload the file.

2. Delegate is nothing but the data user who tries to access the data and download that after getting access permission from private cloud.

3. Cipher Text classes is nothing but data stored in a encrypted manner in live cloud i.e.., Drive HQ server.
4. Access Privileges is given by Private Cloud for the registered users at the time of registration and also based on the user request.

- Setup$(1^\lambda, n)$: executed by the data owner to setup an account on an *untrusted* server. On input a security level parameter 1^λ and the number of ciphertext classes n (i.e., class index should be an integer bounded by 1 and n), it outputs the public system parameter `param`, which is omitted from the input of the other algorithms for brevity.
- KeyGen: executed by the data owner to randomly generate a public/master-secret key pair $(\mathtt{pk}, \mathtt{msk})$.
- Encrypt(pk, i, m): executed by anyone who wants to encrypt data. On input a public-key `pk`, an index i denoting the ciphertext class, and a message m, it outputs a ciphertext $\mathcal{C}$.
- Extract$(\mathtt{msk}, \mathcal{S})$: executed by the data owner for delegating the decrypting power for a certain set of ciphertext classes to a delegatee. On input the master-secret key `msk` and a set $\mathcal{S}$ of indices corresponding to different classes, it outputs the aggregate key for set $\mathcal{S}$ denoted by $K_{\mathcal{S}}$.
- Decrypt$(K_{\mathcal{S}}, \mathcal{S}, i, \mathcal{C})$: executed by a delegatee who received an aggregate key $K_{\mathcal{S}}$ generated by Extract. On input $K_{\mathcal{S}}$, the set $\mathcal{S}$, an index i denoting the ciphertext class the ciphertext $\mathcal{C}$ belongs to, and $\mathcal{C}$, it outputs the decrypted result m if $i \in \mathcal{S}$.

PROCEDURAL EXPLANATION

STEP 1

In the initial step, the data owner should set up a system for granting data storage and also allowing the data users to access the data based on the privileges what they have. As we all know that the data is mainly uploaded by data owner into a semi trusted cloud server, we need to generate a public and master–secret key for the file that was uploaded by the owner, so for that the plain text in the file into encrypted manner.

STEP 2

In the second step the data owner will generate a public key called as identity for each and every individual file that was uploaded in the cloud with the help of token which was given by the cloud for that data owner at the time of registration. Here the private cloud try to generate a master key input for the set of data users whom he wish to give data access and he will restrict the access of un-authorized users whom he don't like to give access.

STEP 3

In the step3 the data which is uploaded in the cloud server will be converted into encrypted manner by using the constraints public key, messages and as well as index for whom the data should be decrypted and viewed (i.e. here index means the data user who is having permission from data owner and in turn from private cloud can only access the data, for others this index cannot be allow for accessing the data).

STEP 4

In this phase the data users who has the ability to access the files which was uploaded in the storage node can extract the key which was generated by substituting their login details along with the secret key that was generated by private cloud while storing the file into the data base.

STEP 5

This is the last step for this application where the data is decrypted and can be viewed by the data user who substitutes all the details correctly. If the data user fails to substitute any of the following identity correctly, he can't able to access the data in plain text manner. And if the data user who try to view the file details in plain manner need to have permission from the private cloud account ,if not he can't able to view the data in a plain text manner.

V.IMPLEMENTATION STAGE

Implementation is the stage where the theoretical design is automatically converted into programmatic manner. In this application we have totally 6 modules and now let us discuss about those 6 modules in detail as follows:

1) SYSTEM CONSTRUCTION MODULE

In this module we will try to construct a system with following attributes like: Single cloud data storage (i.e. Drive HQ), Single Admin, Single Private Cloud, Multiple data users and data owners account where all these roles combine called to be as a system. And for our project we are taking DriveHQ as the storage medium for storing the data lively into the cloud instead of storing them partially into the local database.

2) DATA USER /OWNER REGISTRATION MODULE

Initially the data user or data owner need to be registered into the application for storing their valuable data and also for accessing the data to and from the cloud. So for getting the data access the user/owner need to be registered first. Once they get registered then only they can able to enter into their accounts and store their valuable data without duplication.

3) TOKEN GENERATION MODULE

Once the data user or data owner is registered into the application. The user need to get token from the private cloud in order to enter into his account and try to do various operations inside the application. So this token generation module is mainly operated by the private cloud and once the private cloud issues the token then only the user can login into their accounts and do the transactions if not they can't able to enter into their accounts.

4) DATA ENCRYPTION AND DECRYPTION MODULE

This is the module where the data owners try to upload the data into the cloud server; he will try to encrypt the data before it gets stored into the live cloud server like DriveHQ. Once the data owner encrypts the data with the help of AES algorithm, the same data can be accessed in plain text if the same user who tries to obtain the access privileges from the private cloud. As here in this application, I was using the AES, it is a secret key encryption algorithm which doesn't require the user substitution at the time of encryption or decryption. This will be done automatically by the cloud server who is granting data storage and access privileges for the corresponding users.

5) MONITORING OF USER ACTIVITIES

This module is an extension for our current application, where the current cloud servers there is no facility to monitor the user activities like upload, download or update of data which is done by the user at the time of data accessing to and from the cloud. So in this application we kept a module like monitoring of user activities where the user activities can be monitored by the admin who is available in our application. Here the admin can view the data and time of the user operation what he performed in this application.

6) ACCESS PRIVILEGE MODULE

This is a module in which the private cloud has the permission to grant access or privileges for the data users at the time of registration or based on user request. Whenever a user is registered for the first time he needs to get token key from the private cloud. At that time only the private cloud will give access privilege for the registered users(i.e. Like Semi/No access or /Full Access).If the private cloud give all the three access like upload & download & update, then such an access is known as full access. If the same private cloud gives the partial access like any one or two access among the set, then they are known as semi access. If the same private cloud doesn't wish to give any access from his side to the user, then he will be treated as a No Access user. By default the user has a privilege like read access at the time of account creation.

VI. CONCLUSION

In this paper, we finally designed and developed a novel cloud server with advanced facilities like encryption of data which is stored into the cloud server and also an advanced feature like information de-duplication, in which no duplicate data is stored inside the cloud. For this we have used a live cloud service like drivehq cloud service with a public cloud access account in order to prove these advanced features that are proposed in this current paper. Our proposed system uses this new de-duplication technique in order to main concept for resource allocations. As a proof of concept, we implemented a prototype of our proposed authorized duplicate check scheme and conduct test bed experiments on our prototype. We showed that our authorized duplicate check scheme incurs minimal overhead compared to convergent encryption and network transfer.

VII. REFERENCES

[1] R.A. Popa, C.M.S. Redfield, N. Zeldovich, and H. Balakrishnan,―CryptDB: protective Confidentiality with Encrypted QueryProcessing,‖ Proc. twenty third ACM Symp. operative Systems Principles,Oct. 2011.

[2] S. Quinlan and S. Dorward. Venti: a new approach to archival storage. In *Proc. USENIX FAST*, Jan 2002.

[3] J. Xu, E.-C. Chang, and J. Zhou. Weak leakage-resilient client-side deduplication of encrypted data in cloud storage. In *ASIACCS*, pages 195–206, 2013.

[4] J. Yuan and S. Yu. Secure and constant cost public cloud storage auditing with deduplication. *IACR Cryptology ePrint Archive*, 2013:149, 2013.

[5] K. Zhang, X. Zhou, Y. Chen, X.Wang, and Y. Ruan. Sedic: privacyaware data intensive computing on hybrid clouds. In *Proceedings of the 18th ACM conference on Computer and communications security*, CCS'11, pages 515–526, New York, NY, USA, 2011. ACM

[6] Deduplication + Amazon S3 will save you t]ime and money. White Paper: Published June 2008 data United States Patent 5,813,008; Benson, et al., September 22, 1998 at US Pat. Ofc.

[7] M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. *J. Cryptology*,22(1):1–61, 2009.

[8] *"Single Instance Storage in Windows 2000" (PDF). Microsoft Research and Balder Technology Group. August 2000.*

[9] S. Quinlan and S. Dorward. Venti: a new approach to archival storage. In *Proc. USENIX FAST*, Jan 2002.

[10] D. Ferraiolo and R. Kuhn. Role-based access controls. In *15th NIST-NCSC National Computer Security Conf.*, 1992.

[11] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. *IEEE Computer*, 29:38–47, Feb 1996.

[12] J. Li, X. Chen, M. Li, J. Li, P. Lee, andW. Lou. Secure deduplication with efficient and reliable convergent key management. In *IEEE Transactions on Parallel and Distributed Systems*, 2013.

[13] W. Jansen and T. Grance, ―Guidelines on Security and Privacy inPublic Cloud Computing,‖ Technical Report Special Publication800-144, NIST, 2011.

[14] M. Armbrust et al., ―A read of Cloud Computing,‖ Comm. of theACM, vol. 53, no. 4, pp. 50-58, 2010.

[15] H. Hacigu¨mu¨ s¸, B. Iyer, C. Li, and S. Mehrotra, ―ExecutingSQL over Encrypted knowledge within the Database-Service-ProviderModel,‖ Proc. ACM SIGMOD Int'l Conf. Management knowledge, June2002.

VIII. ABOUT THE AUTHORS

BELLALA ANUSHA is currently pursuing her 2 years M.Tech in Department of Computer Science and Engineering at Pydah College of Engineering and Technology, affiliated to JNTUK University, AP, India. Her area of interest includes Networks and Cloud Computing.

V.SRIKANTH completed his MCA and M.Tech. in Computer Science and Engineering. He is currently working as Assistant Professor of Department of Computer Science and Engineering at Pydah College of Engineering and Technology, affiliated to JNTUK University. He is having teaching experience of more than 13 years. His areas of interest include Computer Networks, Data Mining and Cloud Computing.