

PERFORMANCE EVALUATION OF A MULTI CLOUD DATA STORAGE WITH RANKED SEARCH

RAJABABU PUKKALLA^{#1}, K.VENKATA RAO^{#2}

^{#1} M.Tech Scholar, Department of Computer Science and Systems Engineering,
College of Engineering, Andhra University, Visakhapatnam, AP, India.

^{#2} Professor, Department of Computer Science and Systems Engineering,
College of Engineering, Andhra University, Visakhapatnam, AP, India

ABSTRACT

In current day's 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. And another feature that was not available in the current cloud systems is all the search which is done in the cloud will be of single keyword based but not of multi keyword, so this leads a great difficulty for the user, if he/she forget the file name what they gave for that stored file at the time of retrieving. Till now there was no mechanism available to store the data in a encrypted manner in all public clouds and even private clouds. In this paper, we have introduced an concept like encryption of data before it is stored into the live cloud server like (DRIVEHQ) and also we proposed a new scheme to deal with Privacy preserving Ranked Multi-keyword Search in a Multi-owner model (PRMSM). In order to give rank for the search files which was asked by the user and preserve the privacy of relevant scores not to be revealed for un-authorized users, we have used a novel Additive Order(AO) and Privacy Preserving Function family. In this paper as an extension we have implemented a new concept like advanced authorization of cloud users, where the cloud user registration either owner or user need to get activation permission from the cloud server. The user who got the activate permission will receive the login password for their registered mail id, with that only the user or owner can login, if not login fails this give more security for the current application. By conducting various experiments on our proposed algorithm by taking DRIVEHQ as real cloud service, we finally came to a conclusion that this is the first time to implement such a function which gives high level of security for data during insertion and retrieval compared to various primitive clouds.

Key Words: Encryption, Multi Keyword Search, Authorization, Sub-Dictionaries.

I. INTRODUCTION

In current days there was a huge demand for the cloud computing as a lot of companies try to store all their valuable data on remote servers rather than on local machines. Hence cloud server acts as a main source storing and retrieving data from a remote machines. As there was a huge demand for the cloud computing domain as all the information is usually processed remotely in unknown machines those users do not own or operate not on their local systems or local PCs. Even though cloud has gained a lot of user's attention in storing their valuable data inside that memory blocks, it failed to give high level of security for the data which is stored. This is mainly due to the problem like no appropriate encryption technique is available in the current cloud server while storing the data inside the memory block. In the cloud server, there will be two types of users like data owners and data users, where the data owner is the person who will upload the sensitive or valuable data into the cloud server location and now the data users are the persons who will try to access the data, which is uploaded by the single or multiple data owners. Now a days there was no proper mechanism like encryption of data which is stored into the cloud, so all the data which is stored in the cloud will be stored in a plain manner, which is the main limitation for data integrity. Next in the current cloud service providers there is no concept like ranking the files which is stored and uploaded by the data owners. In the cloud there are various types of services available in which Data Base as a Service (DaaS) is one of the main and prominent services among others. This service is not having security for the data which is stored in the cloud, compared with various other cloud services, hence our main motto is to provide security for this DaaS service by integrating various encryption and other techniques are proposed in this current paper.

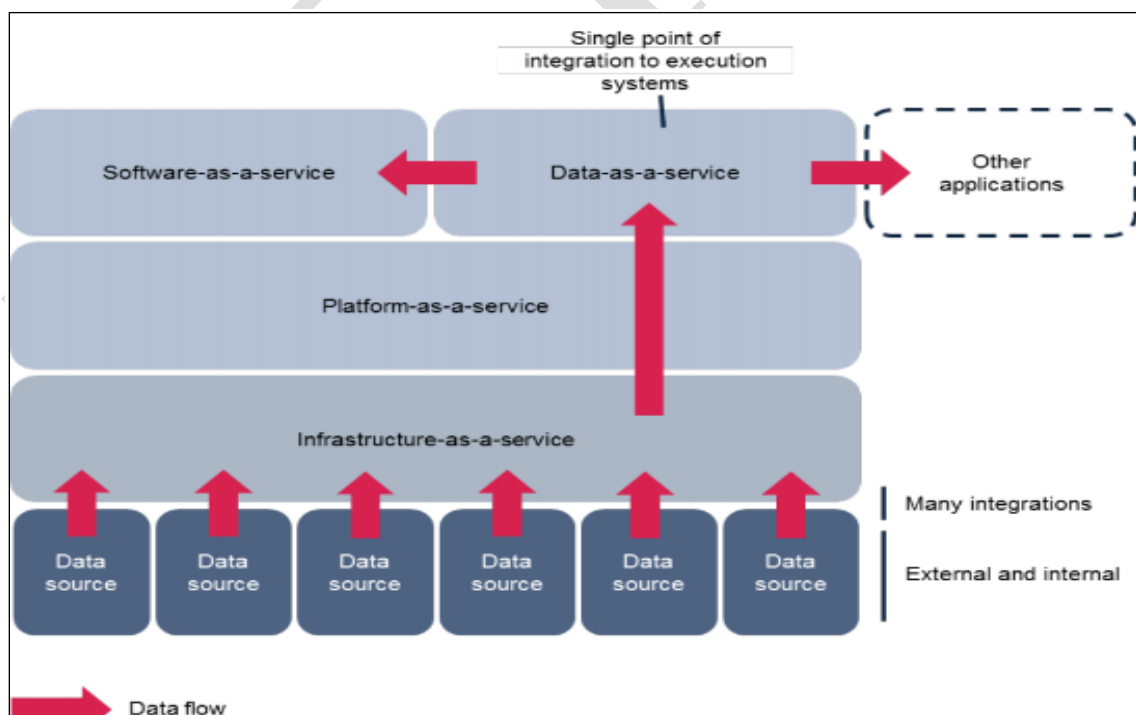


Figure.1. Represents various types of Cloud Services including new service like Database as a Service (DaaS)

From the above figure 1, we can clearly find out that there are four different services available and one among them is DaaS, which is the main service what we are using now for providing security for that and prove that this service also gives the best security for the data which is stored inside the cloud memory locations [1],[2]. Now let us discuss about each and every service in detail as follows:

- 1) IaaS (Infrastructure as a Service)
- 2) PaaS(Platform as a Service)
- 3) SaaS(Software as a Service)
- 4) DaaS (Data /Data Base as a Service)

Now let us discuss about each service in detail and find out the minor similarities that are available between each and every service.

1) IaaS (Infrastructure as a Service)

In this service the cloud server mainly deals with application level and it is basically used to set the platform for the users. The main persons who come under this service is IT Professionals, this is clearly shown in the figure 2.

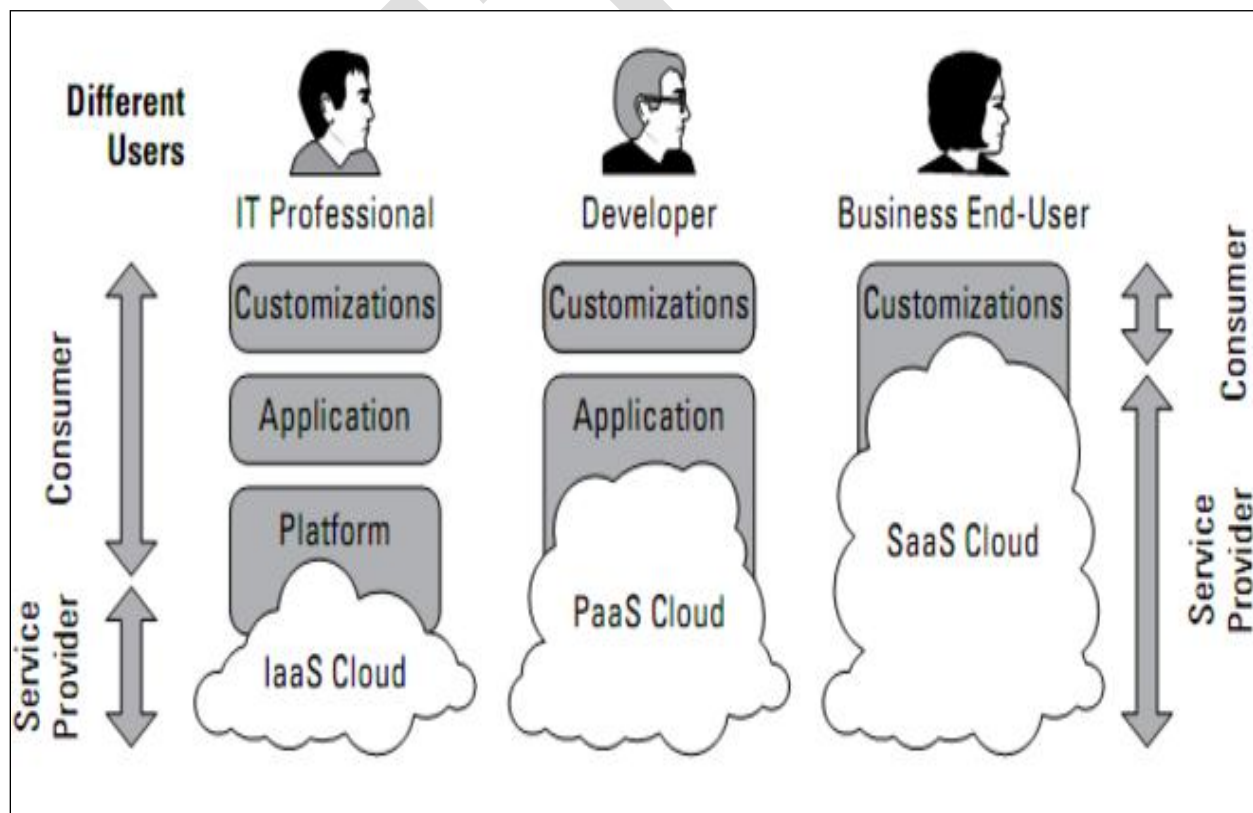


Figure.2. Represents Various Role Based Cloud Service Providers with Their Individual Functionality

2) PaaS (Platform as a Service)

The second and one of the most important service in cloud computing is Platform as a Service, where this is mainly used for customization of cloud server, where the developer comes under this service. Here the cloud server customizes which type of platforms are needed for their company usage are seen in this service.

3) SaaS (Software as a Service)

The third and one of the best services in cloud computing is Software as a Service, where this is mainly used for a consumer to use the cloud service provider's applications running on a cloud IaaS. Generally business end-users come under this service where all the software's that are required for running the cloud are processed in this service.

4) DaaS (Data/Database as a Service)

The last and one of the new services that was launched and included in various cloud client services is DaaS, which is clearly seen in figure 2. This DaaS service is used mainly for storing the data base, tables and data in the form of fragments and packets [3],[4],[5]. As this is having various advantages compared with other cloud client services, it has a small limitation like the data which is stored in this DaaS is not stored in the encrypted manner which is stored in the plain manner.

II. RELATED WORK

In this section we will mainly discuss about the new concept like multi keyword which was introduced in this paper and also about ranking of files. Now let us look at them in details:

ABOUT MULTI KEYWORD SEARCH

In the current cloud servers, we found that there was no facility like multi keyword search over an encrypted cloud data. Now a day's almost all cloud servers utilize single keyword search with only one keyword like filename and there is no concept like searching the same file with multiple attributes. In the current cloud servers if the user who wishes to download any file from the cloud server or to get permission from the cloud owner, he needs to give the exact name of the file correctly in the search bar, then only he can be able to search that file. If the user forget that file name during his search process he can't able to download the exact file from the owner until he substitutes that file name correctly by re-collecting that file information. So this is a one of the major problem in almost all cloud service providers [6]. So in this paper we have implemented a new keyword search like multi keyword search, in which the data owner while uploading the data in the current cloud enters a file name along with a sub-item like keyword and then he browse the file from the desired location. So in this way he is giving multi inputs while

he uploads the sensitive data. So if the user who wishes to download the file from the cloud, he/she can either enter file name or keyword whatever they feel better during search. For example: Data Owner wants to upload a file like Jsp which consists the data about Java Server Pages. Now he will give the file name like Jsp.txt or some other filename like divya.txt and he then give the keyword as Java, whatever he feel better. So once he uploads the file in the above manner, the file will be uploaded at the end. So when a data user who wish to download that file, he can either give the search input as filename or either keyword or both. So in this way multi keyword will give enhance facility for searching and retrieving the files from the cloud servers[7].

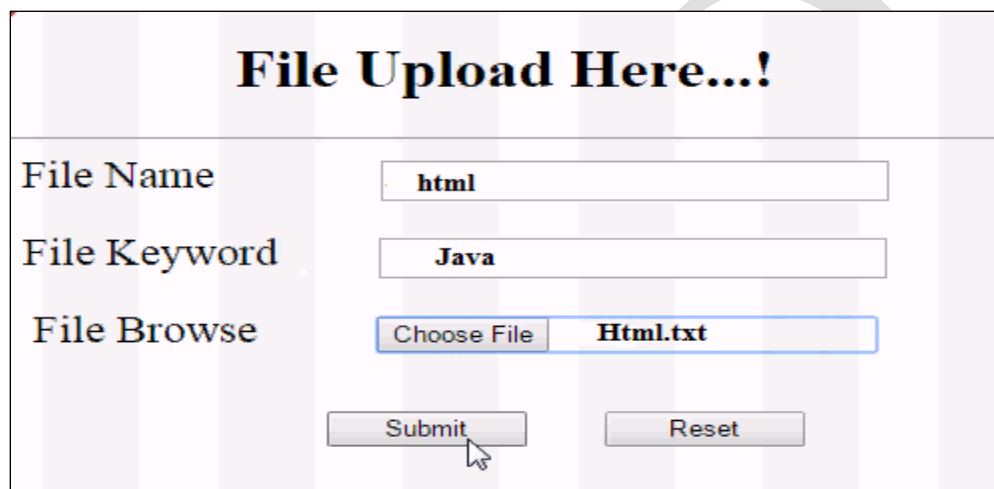


Figure 3. Represents the File Uploaded by a Data Owner in a Multi Keyword Process

From the figure 3, we can clearly get an idea that multi keyword search consists of multiple attributes like filename, keyword and browse option to choose a file. Here we choose a filename as Html.txt and we gave the keyword name as Java and finally the data owner browse the file from any of the Drive location of that PC and uploaded in the cloud. So if the data user who wishes to download the data can either enter filename or keyword name or both for retrieving the file from the encrypted cloud storage area. The above figure clearly justifies that multi keyword gives a better level of data retrieval for the end users.

III. PRIVACY PRESERVING RANKED MULTI-KEYWORD SEARCH IN A MULTI-OWNER MODEL (PRMSM)

In this section we will find out the proposed system and its architecture that was used in the current paper. In this paper we have implemented privacy preserving multi keyword search in a multi owner model also known as PRMSM. Now let us discuss about this in detail as follows:

Main Entities

For any type of cloud applications, there may be several cloud service providers but for all types of CSP, maximum they contain these four main entities

- A. Data Owner Entity,
- B. Admin Entity
- C. Data User/Search User Entity
- D. Cloud Server Entity

The data owner entity is the starting entity which is defined as a person who may be an individual or sometimes an enterprise, who wishes to outsource a collection of documents $D = (D1, D2, \dots, Dn)$ in encrypted form $C = (C1, C2, \dots, Cn)$ to the cloud server and still preserve the search functionality on outsourced data. Here we assume that documents are labeled with D and if there are many documents to be out sourced they are represented as $D1, D2$ and so on. Here in our proposed application, we take sample text documents as input where initially all the text documents are of plain text and our main motto is to store them in a secure manner inside a cloud server. Once the text files which consists the sensitive data are encrypted and then they are stored into the cloud server, they are termed as $C1, C2$ and so as they were encrypted by the data owner at his level before out sourcing into the Admin.

Here once the admin after login into his account, he has the facility to receive the data request which is send to that by data owner after an initial encryption. This admin will now receive all the files which was uploaded by various data owners and then they will be re-encrypted by the admin at his level and then it was send to the cloud server. During this stage the re-encrypt method will encrypt not only the file content but also the file details like file name, file upload date and time and so on. This double encryption or re-encrypt gives much more security for our proposed application compared with various primitive cloud service providers. Now the data which is uploaded by admin will be reached to the cloud server, where the cloud server is an important entity among all the four as this is the only entity which has the capability to store the encrypted documents into its storage area. Once the cloud owner encrypts the text documents and it is uploaded [9]- [12], then immediately they will be received by the cloud server and it will then store in its storage area securely. When a search user try to download any file, he will send the input as either filename or file keyword so that immediately the file request will be identified by the data base records and if the input keyword is matched the file will be downloaded and if that was not matched it will be identified as data not found. During this process if any intruder try to access illegally the data by substituting the others identities, he will be identified as a trapdoor user and file can't be downloaded.

A search user is the last entity who wishes to download the files from the cloud server by giving valid inputs for searching the files and then download those files in a secure manner. The search user has following three steps to be performed for downloading the Encrypted text documents from the cloud server, they are as follows: First, the search user initially after registration, he will be login into his account by substituting all the valid details what he stored during registration. Second, according to the search keywords, the search user uses the same secret key along with any of the search parameter to generate a decryption key and sends it to the cloud server. Once if the input parameters along with secret key are matched with server records, then the search user receives the matching document collection from the cloud server and decrypts them with the symmetric key which is dynamically sent to the search user mail id at the time of downloading the file. If the user substitutes valid decryption key can only download the file in a plain manner or else the file will be in encrypted manner.

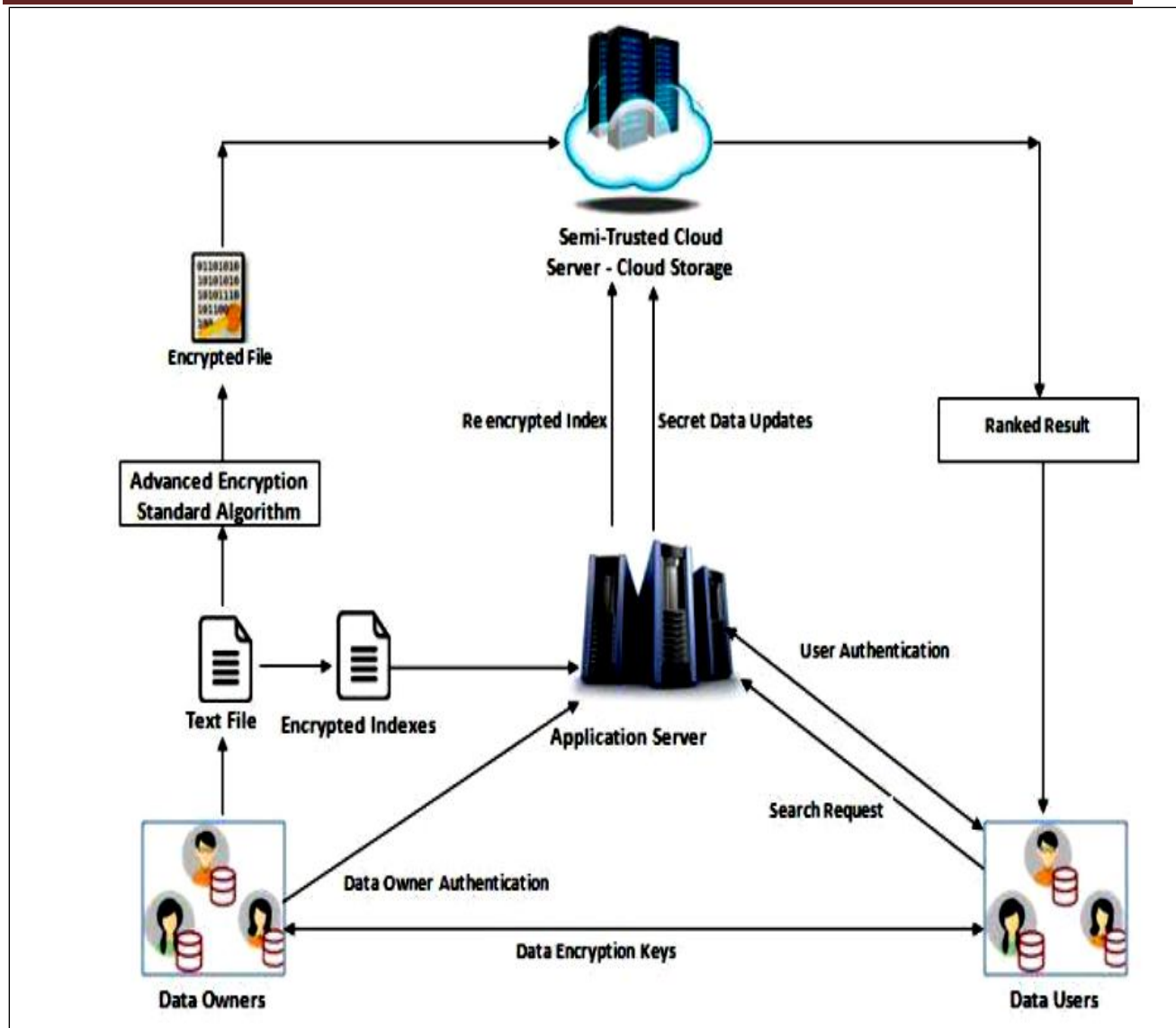


Figure 4. Represents the Privacy Preserving Ranked Multi-keyword Search in a multi-owner Model (PRMSM)

From the above figure 4, we can clearly represent the proposed architecture flow diagram of our current paper, where it contains totally three roles like Data owner, Cloud Server, Admin Server and Search users. Here each and every one has individual roles and all the access in this current model is in form of De-Centralized manner. In the primitive or existing cloud servers, the data access will be obviously in a centralized manner, where the data which is uploaded by owner will be stored inside the cloud server and in turn the access will be in the hands of server itself, But there was no single access control for the owner or user in the current cloud service providers. So in this paper we for the first time implemented architecture like De-Centralized access by giving individual rights for each and every individual. The data which is uploaded by the data owner is having a right to give access or deny the access of his uploaded file at the time

of user search request. Here the owner will receive all the search requests done by various data users or search users within the cloud and once if the data owner really wish to give access to user then only he will click on allow button so that access will be granted and symmetric key as a decryption key will be send for the requested search user, if not access will be restricted by the owner and he will be treated him as a trapdoor user. Here the Cloud server has a capability to receive all the user requests and in turn send that request to the appropriate data owners who uploaded the data into the cloud. Here the data which is uploaded is in form of encrypted manner and the records are almost text documents with a valid sensitive data and they are stored in a secure manner onto the cloud storage area.

Assumptions

Let $D = (D_1, D_2, \dots, D_n)$ be a set of documents and

$K = (k_1, k_2, \dots, k_m)$ be the dictionary consisting of unique keywords in all documents in

D , where $\forall i \in [1, m] \quad k_i \in \{0, 1\}^*$.

$C = \{C_1, C_2, \dots, C_n\}$ is an encrypted document collection stored in the cloud server.

I_i is a searchable index associated with the corresponding encrypted document C_i .

If A is an algorithm then $a \leftarrow A(\dots)$ represents the result of applying the algorithm A to given arguments.

Let R be an operational ring, we write vectors in bold, e.g. $\mathbf{v} \in R$.

The notation $\mathbf{v}[i]$ refers to the i -th coefficient of \mathbf{v} .

We denote the dot product of $\mathbf{u}, \mathbf{v} \in R$ as

$$\mathbf{u} \otimes \mathbf{v} = \sum_{i=1}^P \mathbf{u}[i] \cdot \mathbf{v}[i] \in R.$$

We use $\lfloor x \rfloor$ to indicate rounding x to the nearest integer, and $\lfloor x \rfloor, \lceil x \rceil$ (for $x > 0$) to indicate rounding down or up.

Here in the current application we denote the function $C_i = E_S[D_i]$ is the encrypted version of the document D_i , which is mainly computed by using a semantically secure encryption scheme E with a secret key S . To enable multi-keyword ranked search capability, the data owner always constructs a searchable index termed as “ I ” that is built on “ m ” distinct keywords $K = (k_1, k_2, \dots, k_m)$ extracted from the original dataset D . Both I and C are outsourced to the cloud server. To securely search the document collection for one or more keywords $K^- \in K$, the authorized data user uses search trapdoor (distributed by the data owner) that generates the search request to the cloud server. Once the cloud server receives such request, it performs a search based on the stored index I and returns a ranked list of encrypted documents

$L \subseteq C$ to the data user. The data user then uses the secret key S , securely obtained from the data owner, to decrypt received documents L to original view.

IV. ADDITIVE ORDER AND PRIVACY PRESERVING FUNCTION

In this section we will find the notations and an equation to find the ranking for the documents which was uploaded into the cloud by the data owner. This is done with the help of additive order and privacy preserving function. Generally in number theory, an **additive function** is defined as an arithmetic function termed as $f(n)$ of the positive integer say ' n ' such that whenever a and b are prime, the additive function is defined as the summation of all the co prime values. This is represented as follows:

$$F(ab) = f(a) + f(b).$$

In this paper we use the sum of the relevance scores as the metric to rank search results. Here we introduced various encoded strategies for ranking the relevance scores. Initially, the cloud server computes

$$V_{i,j} = \sum_{t \in W_f} V_{i,j,t}$$

Now to find the ranking for the search values, the sum of all relevance scores between the j th file and matched keywords for O_i , and the auxiliary value

$$T_{i,j}(y) = \sum_{t \in W_f} T_{i,j,t}(y).$$

The relevance score between a keyword (W) and a document (F) represents the frequency or count in which that the keyword appears in the document. It can be used in searchable encryption for returning ranked results. A prevalent metric for evaluating the relevance score is $TF \times IDF$, where TF (term frequency) represents the frequency of a given keyword in a document and IDF (inverse document frequency) represents the importance of keyword within the whole document collection. Without loss of generality, we select a widely used expression in [13] to evaluate the relevance score as

$$Score(\widetilde{W}, F_j) = \sum_{w \in \widetilde{W}} \frac{1}{|F_j|} \cdot (1 + \ln f_{j,w}) \cdot \ln(1 + \frac{N}{f_w})$$

Where

$F_{j,w}$ denotes the TF of keyword w in document F_j ;

F_w denotes the number of documents contain keyword w ;

N denotes the number of documents in the collection; and

$|F_j|$ denotes the length of F_j , obtained by counting the number of indexed keywords.

V. IMPLEMENTATION PHASE

Implementation is the stage where the theoretical design is converted into programmatically manner. In this stage we will divide the application into a number of modules and then coded for deployment. We have implemented the proposed concept on Java programming language with JEE as the chosen language in order to show the performance this proposed Privacy preserving Ranked Multi-keyword Search in a Multi-owner model (PRMSM). The front end of the application takes JSP, HTML and Java Beans and as a Back-End Data base we took My SQL data base along with a Real Cloud Service provider called as DRIVEHQ Cloud Service provider. This cloud service provider will provide a space up to 2 GB for storing the files which is used by the application. The application is divided mainly into following 4 modules. They are as follows:

1. System Model Module
2. Data User Authentication Module
3. Illegal Search Detection Module
4. Search Over Multi Owner Module

1. System Model Module

In the first module, we develop the System Model to implement our proposed system. Our System model consists of Admin, users, data owners, and Cloud Servers. Admin provides the accessibility to Data-owners. Initially Data-owner needs to register and admin approves the each data owner request. The respective Password and login credentials will be sent to the Email ID of Data owner. In Users sub-module, each user has a global identity in the system. A user may be entitled a set of attributes which may come from multiple attribute authorities. The user will receive a secret key associated with its attributes entitled by the corresponding attribute authorities. In data owner's sub-module, the proposed scheme should allow new data owners to enter this system without affecting other data owners or data users, i.e., the scheme should support data owner scalability in a plug-and-play model. In Cloud Server sub-module of system model, the owner sends the encrypted data to the cloud server through Admin. They do not rely on the server to do data access control. But, the access control happens inside the cryptography. That is only when the user's attributes satisfy the access policy defined in the cipher text; the user is able to decrypt the cipher text. Thus, users with different attributes can decrypt different number of content keys and thus obtain different granularities of information from the same data.

2. Data User Authentication Module

To prevent attackers from pretending to be legal data users performing searches and launching statistical attacks based on the search result, data users must be authenticated before the administration server re-encrypts trapdoors for data users. Traditional authentication methods often follow three steps. First, data requester and data authenticator share a secret key, say, k_0 . Second, the requester encrypts his personally identifiable information d_0 using k_0 and sends the encrypted data (d_0) k_0 to the authenticator. Third, the authenticator decrypts the received data with k_0 and authenticates the decrypted data. The key point of a successful authentication is to provide both the dynamically changing secret keys and the historical data of the corresponding data user.

3. Illegal Search Detection Module

In this proposed scheme, the authentication process is protected by the dynamic secret key and the historical information. We assume that an attacker has successfully eavesdropped the secret key. Then he has to construct the authentication data; if the attacker has not successfully eavesdropped the historical data, e.g., the request counter, the last request time, he cannot construct the correct authentication data. Therefore this illegal action will soon be detected by the administration server. Further, if the attacker has successfully eavesdropped all data of U_j , the attacker can correctly construct the authentication data and pretend himself to be U_j without being detected by the administration server. However, once the legal data user U_j performs his search, since the secret key on the administration server side has changed, there will be contradictory secret keys between the administration server and the legal data user. Therefore, the data user and administration server will soon detect this illegal action.

4. Search over Multi Owner Module

The proposed scheme should allow multi-keyword search over encrypted files which would be encrypted with different keys for different data owners. It also needs to allow the cloud server to rank the search results among different data owners and return the top- k results. The cloud server stores all encrypted files and keywords of different data owners. The administration server will also store a secret data on the cloud server. Upon receiving a query request, the cloud will search over the data of all these data owners. The cloud processes the search request in two steps. First, the cloud matches the queried keywords from all keywords stored on it, and it gets a candidate file set. Second, the cloud ranks files in the candidate file set and finds the most top- k relevant files. Finally, we apply the proposed scheme to encode the relevance scores and obtain the top- k search results.

VI. CONCLUSION

In this paper, we for the first time implemented a secure multi-keyword search for multiple data owners and multiple data users in the cloud computing environment. In the current cloud servers, there was no concept like encryption of data before it is stored into the cloud. As the encryption was not available in the current clouds, all the data which is stored into the cloud has no security and any one can access that data freely without any restrictions. So in this paper for the first time we have implemented a new concept like encryption of data before it is stored inside the live cloud. Also we have implemented a new concept called as multi keyword search, where the data which is uploaded by the data owner will provide multiple attributes for each and every file during the upload, so this multiple attributes act like a multiple keywords for accessing the file during download or search. To enable the cloud server to perform secure search among multiple owners' data encrypted with different secret keys, we systematically construct a novel secure search protocol. To rank the search results and preserve the privacy of relevance scores between keywords and files, we propose a novel Additive Order and Privacy Preserving Function family. By conducting various experiments on our proposed model, we finally came to a conclusion that this proposed mechanism gives high level of security in terms of data during storage and retrieval.

VII. REFERENCES

- [1] Peter Mell and Timothy Grance (September 2011). The NIST Definition of Cloud Computing (Technical report). National Institute of Standards and Technology: U.S. Department of Commerce. doi:10.6028/NIST.SP.800-145. Special publication 800-145.
- [2] Alcaraz Calero, Jose M.; König, Benjamin; Kirschnick, Johannes (2012). "Cross-Layer Monitoring in Cloud Computing". In Rashvand, Habib F.; Kavian, Yousef S. Using Cross-Layer Techniques for Communication Systems. Premier reference source. IGI Global. p. 329. ISBN 978-1-4666-0961-7. Retrieved 2015-07-29. Cloud Computing provides services on a stack composed of three service layers (Hurwitz, Bloor, Kaufman, & Halper, 2009): Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).
- [3] Gartner. "Gartner IT Glossary". Retrieved 6 July 2015.
- [4] Gartner; Massimo Pezzini; Paolo Malinverno; Eric Thoo. "Gartner Reference Model for Integration PaaS". Retrieved 16 January 2013.
- [5] Loraine Lawson. "IT Business Edge". Retrieved 6 July 2015.
- [6] C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, "Toward Secure and Dependable Storage Services in Cloud Computing," IEEE Trans. Services Computing, vol. 5, no. 2, pp. 220-232, Apr.-June 2012.
- [7] Y.-C. Chang and M. Mitzenmacher, "Privacy Preserving Keyword Searches on Remote Encrypted Data," Proc. Third Int'l Conf. Applied Cryptography and Network Security, 2005.
- [8] <http://www.merriam-webster.com/dictionary/ranking>.
- [9] Towell, G.G. and Shavlik, J.W. (1993), 'Extracting refined rules from knowledge-based neural networks' physicist. Learn, Vol.13, pp.71-101.
- [10] Ehrlich, Melanie; Gama-Sosa, Miguel A.; Huang, Lan-Hsiang; Midgett, Rose Marie; Kuo, Kenneth C.; McCune, Roy A.; Gehrke, Charles (1982). "Amount and distribution of 5-methylcytosine in human DNA from different types of tissues or cells". Nucleic Acids Research.
- [11] Moréra, Solange; Larivière, Laurent; Kurzeck, Jürgen; Aschke-Sonnenborn, Ursula; Freemont, Paul S; Janin, Joël; Rüger, Wolfgang (August 2001). "High resolution crystal structures of T4 phage β -glucosyltransferase: induced fit and effect of substrate and metal binding". Journal of Molecular Biology.
- [12] 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

[13] Erdős, P., and M. Kac. On the Gaussian Law of Errors in the Theory of Additive Functions. Proc Natl Acad Sci USA. 1939 April; 25(4): 206–207.



VIII .ABOUT THE AUTHORS

RAJABABU PUKKALLA is currently pursuing his 2 Years M.Tech in Computer Science and Systems Engineering, College of Engineering, Andhra University, Visakhapatnam, AP, India. His area of interests includes Networks.



K.VENKATA RAO is currently working as an Professor, in Computer Science and Systems Engineering, College of Engineering, Andhra University, Visakhapatnam, AP, India. He has more than 15 years of experience in teaching field. His research interest includes Image Processing.