A Secure Privacy Preserving Information Retrieval Model in Cloud Computing

Saripalli Vinod Manikanta[1], Kondapalli Varaprasad[2]
M.Tech (Software Engg)[1], M.Tech (Asst Prof, Computer Science)[2]

Abstract

Information retrieval over cloud resources or data storage is an interesting research issue with multiple data owners. Privacy preserving data is major concern while transmission data from the clients and data sources. Tree based security models are complex to implement and takes more navigation time when data is huge. We propose a novel privacy-preserving model of base table generation to improve the search implementation and user search performance or response time. Cloud based cache implementation improves the performance without accessing the data from data sources when it is available at midlevel .Our proposed model gives more efficient results than traditional models.

I.INTRODUCTION

The expansion of little hand-held devices and wireless systems administration empowers portable users to get to their information whenever and from anyplace. For reasons of expense and efficiency, users regularly store their information not all alone machine, yet, on remote servers that may likewise offer better availability. At the point when the server is untrusted, users guarantee the classification of their information by putting away it encoded[1][2].

The main pragmatic answer for the issue of looking encrypted information by keyword. Archives and catchphrases are encrypted[3] in a way that enables the server to figure out which archives contain a specific catchphrase W in the wake of getting from the client a snippet of data called a capacity for keyword W. The capacity for W uncovers just which archives contain catchphrase W and no other data. Without a capacity, the server adapts nothing about encrypted reports.

The systems give provable secrecy to encryption, in the sense that the untrusted server can't master anything about the plaintext given just the ciphertext. The systems give controlled searching, so that the untrusted server can't search for a word without the client's approval. The methods bolster covered up inquiries, with the goal that the client may ask the untrusted server to search for a secrecy word without uncovering the word to the server. The procedures additionally supports question segregation, implying that the untrusted server adapts nothing more than the search result about the plaintext[4].

The present mail servers, for example, IMAP servers, document servers and other information stockpiling servers normally should be completely believed—they approach the information, and thus should be trusted not to uncover it without approval—which presents unwanted security and protection chances in applications. Past work tells the best way to construct encrypted document frameworks furthermore, secure mail servers, yet regularly one must forfeit usefulness to guarantee security. The crucial issue is that moving the calculation to the information stockpiling appears extremely troublesome when the information is encrypted, and numerous calculation issues over encrypted information recently had no handy arrangements[5]

II.RELATED WORK

Assume client Alice wishes to peruse her email on various devices: PC, laptop, pager, and so on. Alice's mail entryway should course email to the proper device dependent on the keywords in the email. For instance, when Bob sends email with the watchword "pressing" the mail is directed to Alice's pager. At the point when Bob sends email with the watchword "lunch" the mail is steered to Alice's work area for perusing later. One anticipates that each email should contain few keywords. For instance, all words on the title just as the sender's email address could be utilized as keywords[6].

The mobile people venture gives this email handling ability. There are issues in privacy for database data. There are two types such as public databases and private databases. There are different methods for solutions. Private databases: In this database a user wants to upload its private data to a third part database and wants to store the data private from the third party database admin. Afterwards, the user can retrieve from the third party database all records that contain a special keyword[7][8].

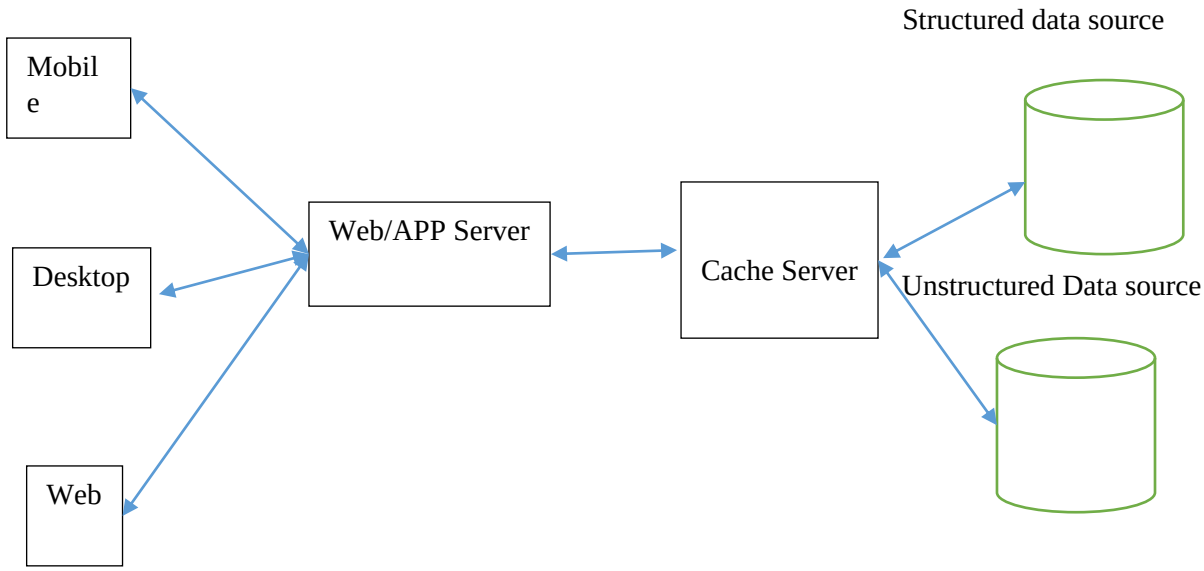
Public Databases are the database data is public but the user is don't know of it and wants to retrieve data or search for data, without knowing to the database admin which data it is. The method is that the user can retrieve the entire database. Public Information Retrieval (PIR) methods allow user to download data from a public database[9][10] with few communication then retrieving the entire

database. We stress however that in every one of these settings the database is public, and the client is endeavouring to recover or discover certain things without uncovering to the database administrator what it is hunting down. In the setting of a solitary public database, it very well may be demonstrated that the database should dependably perform work which is at any rate direct in the span of the database.

III.PROPOSED WORK

We propose an efficient base index model for efficient data retrieval from out sourced or cloud databases .Data, which uploaded over cloud, is encrypted modification of the encrypted key every time when user modify is complex. Usually a key can be constructed through a group key generation protocol whether it is centralized model or distributed model. However, it makes complex if we encode the file with new key when new user added, so we encrypt with old key use this new key for authorization. It minimizes the complexity and improves the results.

For secure information retrieval, data can be encoded after segmentation and uploaded to server and key received from the key generation centre and authentication parameters encoded with file which is uploaded. Searching a keyword over encrypted data components is not possible, so user should be authenticate to use the key and data can be decoded while retrieval from the data sources if not available for cache storage. Key generation protocol is important to generate the key for multiple data owners. key generation centre registers the users at centralized servers and receives the authentication parameters and verifies the authentication with signatures. If user is authenticates, he receives a proof code, end user make a computation and forward the computed value to centralized user, this can be continued to all users and key generation accumulate the results after receiving from all nodes



An experimental model of multi keyword seek in out sourced information bases for information query.There have all the earmarks of being two sorts of systems. One believability is to build up a record that, for each expression of intrigue, records the chronicles that contain. An alternative is to play out a back to back compass without a record. The advantage of utilizing a rundown is that it may be speedier than the back to back channel when the records are gigantic. The impairment of utilizing a document is that putting and overhauling the rundown can be of impressive overhead. So the philosophy of utilizing a document is increasingly sensible for the most part scrutinized just data.

Data owner can save the data in cloud database and does not realize the data is put away in which place. At first we expel the pointless data from the parts and it removes keywords. It scrambles the data utilizing the DES calculation and it gauges the recurrence of the keyword and the table is produced three properties, for example, document id, encoded record and transfer to the server.

Algorithm

1. Upload file to encrypt.
2. System reads file F.
3. Split the file term wise.
4. Enctypt the splited file

5. Identify the file term frequency for the divided blocks.
6. It creates the index table.
7. It uploads to the server.

Keyword	Cipher Keyword	Term Frequency	File ID
Mobile	$%^&*(	4	Abc.html
Apple	*(!!~*^%	3	Hello.docx
Elephant	##||%^$%&	1	Hello.docx
Paper	$%^$%^	2	Main.txt

There seem, by all accounts, to be two sorts of approachs. One believability is to build up a record that, for each expression of intrigue, records the files that contain. A choice is to play out a continuous breadth without a record. The advantage of using a rundown is that it may be speedier than the continuous channel when the records are immense. The burden of using a document is that putting and updating the rundown can be of extensive overhead. So the strategy of using a record is increasingly sensible for by and large examined just data.TF = terms which is present in the document.

IDF= frequency of the keyword in all documents

N_c=Count of file

frequency_Scores[j] = Convert.ToDecimal((1 / termsinfile[j]) * (1 + Math.Log(termfreqs[j])) * Math.Log(1 + (file_count / numberoffiles)));

Web services are service masterminded application which can keeps up business method of reasoning in brought together zone rather than keeping up dealing with module at every individual client end or at UI, it diminishes the overabundance, limit the glitch risks by keeping up the business basis course from the end customers. The guideline favored point of view of the web services are language interoperability, any standard programming language can talk with web service neighbourhood language by using transitional language web service depiction language.

Search Implementation

End client at first registers at data owner to get key which is used to unscramble the figure content in base table, after login end client can be affirmed with client id and key and at whatever point a client progresses a data request, it tends to be changed over to figure question and differences and figure keywords in base table and recoups TF and IDF of figure keyword and procedures archive importance score and time significance score, instead of taking a gander at plain artistic information.

Term frequency based search algorithm:

1. User requests a key and registration.
2. The key is used for authentication and search by the user.
3. User searches for information in the format of plain text.
4. Service verifies the authentication and searches for the query.
5. After processed information search service retrieves for the respective search.
6. Service calculates the relevance count of file.

freq_Scores[j] = Convert.ToDecimal((1 / termsinfile[j]) * (1 + Math.Log(termfreqs[j])) * Math.Log(1 + (file_count / numberoffiles)));

7. Time relevance $T_1,T_2……T_n$ of the occurrence document.
8. For I in freq_scores Begin
9. Relevance_score=frequency_scores[i]*Tr End
10. Format the documents according on relevance score
11. Retrieving the files depends on the file relevance score to user.

IV.CONCLUSION

Usually a request made by the client firs reaches to web or application based on the application type and checks for cache information for required key, because it maintain in the key value pair. If is not available, it redirects to data sources and retrieve the information and forwards to cache server and then send back to the requested user. Cache has a flexibility to expire automatically after expiry time. We can update cache if there is any insert or update on respective collections. Our proposed results more efficient results than traditional models.

REFERENCES

[1] (May 24, 2017). 2016 Monitoring Report on the Data of China's E-Commerce Market [EB/OL]. [Online]. Available: http://www.100ec.
[2] cn/zt/16jcbg/
[3] Amazon. Amazon S3. Accessed: Sep. 5, 2017. [Online]. Available:http://aws.amazon.com/s3/
[4] Windows Azure. Accessed: Sep. 5, 2017. [Online]. Available: http://www.microsoft.com/windowsazure/
[5] Apple iCloud. Accessed: Sep. 5, 2017. [Online]. Available: http://www.icloud.com/
[6] Google App Engine. Accessed: Sep. 5, 2017. [Online]. Available:http://appengine.google.com/
[7] VOLUME 6, 2018 14753
[8] Y.-S. Zhao, Q.-A. Zeng: Secure and Efficient Product Information Retrieval in Cloud Computing

[9] P.Golle, J. Staddon, and B. Waters, ``Secure conjunctive keyword search over encrypted data," in Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.,
[10] 2004, pp. 31_45.
[11] D.X.Song and D. A. Wanger Perrig, ``Practical techniques for searches on encrypted data," in Proc. IEEE Symp. Security Privacy, May 2000, pp. 44_55.
[12] D.Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, ``Public key encryption with keyword search," in Proc. Int. Conf. Theory Appl.Cryptogr. Techn., 2004, pp. 506_522.
[13] H.S.Rhee, J. H. Park, W. Susilo, and D. H. Lee, ``Trapdoor security in a searchable public-key encryption scheme with a designated tester," J. Syst.Softw., vol. 83, no. 5, pp. 763_771, 2010.
[14] K.Ren, C.Wang, and Q.Wang, ``Security challenges for the public cloud," IEEE Internet Comput., vol. 16, no. 1, pp. 69_73, Jan./Feb. 2012.
[15] E.-J. Goh, ``Secure indexes," IACR Cryptol. ePrint Arch., Newark, NJ, USA, Tech. Rep. 1, 2003, p. 216.