Imperial Journal of Interdisciplinary Research (IJIR)
Vol-2, Issue-5, 2016
ISSN: 2454-1362, http://www.onlinejournal.in

Faster Retrieval of Data from Cloud using Feature Extraction

[1]Sandesh Molane, [2]Sachin Mate, [3]Shubham Mahamine,
[4]Rishikesh Nikam & [5]U. A. Mande
[1,2,3,4] Department Of Computer Engineering, Savitribai Phule Pune University, Pune
[5]Professor, Dept. of Computer, Sinhgad College of Engineering, Pune
Maharashtra, India

Abstract*: Cloud computing is currently one of the most sought after fields in the IT industry. Searching the relevant data from cloud has become tedious over the years. This paper deals with reducing the retrieval time by directly searching over encrypted data. This technique not only reduces search time but also does not compromise with security of stored data. There are many existing techniques which use methods like fuzzy logic, ranked search, homomorphism encryption, etc. to search on the encrypted data in cloud. The disadvantage with these methods is that they fetch many documents for the given query at the cost of increased retrieval time. In this paper Advanced Encryption Standard (AES) is used for encryption of both data and queried keywords. While uploading the data Features are extracted from each document. When user fires a query, trapdoor of that query is generated and search is performed by finding the correlation among documents stored on cloud and query keyword, using Pearson Correlation. Also concept of generalized inverted index is used to speed up the search time and performance.*

Keywords*: Cloud computing, AES Encryption, Pearson Correlation, Generalized Inverted Index.*

1. Introduction

This project is basically divided into two parts – User uploading the data and searching the query.

Firstly user uploads the data which is encrypted and stored on Cloud using AES algorithm. The various steps are summarized in following lines. It uses Stop-word bag filter to remove commonly used words. This helps to reduce the file size and to remain focused on main content. After this port stemmer is used to find root words. The next step is bucket construction in which bucket of root words is constructed. The bucket construction requires threshold value which is fixed by programmer. The next step in process is Index construction in which index is assigned to the buckets previously constructed. These indexed buckets are then stored as feature file. These feature files are related to original files being uploaded. These feature files are stored as a tree.

The second part is searching required data in the uploaded files. For this a query is fired and it goes through same process of stop-word removal and port stemming. After this word matrix is created and encrypted using AES algorithm. This encrypted query is called as trapdoor. This trapdoor is correlated with already stored feature file in tree using Pearson Correlation. The Generalized Inverted Index (GINIX) is used to minimized the displayed results.

2. Literature Survey

2.1 AES:

2.1.1 A Framework Based on RSA and AES Encryption Algorithms for Cloud Computing Services:

The entities in this system are: Sender, Receiver and Cloud Storage System (CSS). Sender requests its public key from cloud system. After this the cloud service generates the private key (PK), file identifier (ID), public key (PB) and a bid random number (RB). After this, cloud service sends the created public key and the ID of file to the user. And then, sender sends the encrypted file and its ID to the system, while all the process of sending file is encrypted by RSA algorithm. Encryption by RSA algorithm helps to increase the security of all processes of file transfer, key exchange and security of cloud storage service.

The second part of this system is sending file from cloud storage system to the receiver. For this purpose, receiver sends a request for the list of files and then cloud system will send the list of files to the user. Here, receiver can choose the file he is going to download from the cloud system and after making the download request, the user sends the name of file to cloud system and user sends his public key to the system as well. The use of this public key is to encrypt the secret key of the symmetric encryption algorithm.

The last part is finding the requested file by cloud storage system (CSS) and then encrypting this file using AES encryption algorithm. AES got a secret key which is the RB already generated when the cloud was generating the private key and public key. Then, the cloud storage system encrypts the RB with the public key which the receiver sent it previously and CSS will send both RB and also the requested file to the user. RSA encryption algorithm is used to encrypt the RB.

Encrypting the big random number (RB) makes it impossible for attackers to attack the file while it is transferring to the receiver, and attackers cannot see the content of the sent packet.

2.1.2 Use of Digital Signature with Diffie Hellman Key Exchange and AES Encryption Algorithm to Enhance Data Security in Cloud Computing:

In the proposed architecture, researchers are using three ways protection scheme. Firstly Diffie Hellman algorithm is used to generate keys for key exchange step. Then digital signature is used for authentication, thereafter AES encryption algorithm is used to encrypt or decrypt user's data file. All this is implemented to provide trusted computing environment in order to avoid data modification at the server end. For the same reason two separate servers are maintained, one for encryption process known as (trusted) computing platform and another known as storage server for storing user data file. When a user wants to upload a file to the cloud server, first key are exchanged using Diffie Hellman key exchange at the time of login, then the client is authenticated using digital signature. Finally user's data file is encrypted using AES and only then it is uploaded to another (cloud) Storage server. Now when client is in need of some file, it is to be downloaded from cloud server. For that purpose, when user logs in, first encryption keys are exchanged, file to be downloaded is selected, authentication takes place using digital signature then, AES is used to decrypt the saved file and client can access the file.

2.1.3 Implementation of Data Privacy between Nodes Using AES in Wireless Ad Hoc Networks:
In this paper, the security of data is preserved while it is in ad-hoc transit mode. For this AES-256 is used using the open source software 'AES Crypt' which implements it in Cipher Block Chaining (CBC) mode. This is implemented using C language as it is a compiled language. It means object file is made only once during compilation process. Makefile is used to generate an object file from multiple C and header files. The object file is a binary file that can be directly implemented on a processor. As the AES code has to be used multiple times, an object can be made and reused any number of times. The object file is then executed from the python script whenever a file is needed to be encrypted or decrypted. The key is passed to the object file at the time of execution along with source file name and destination file name.

2.2 PEARSON CORRELATION:

2.2.1 Collaborative Filtering Based Simple Restaurant Recommender:

In this paper the authors have tried use recommend a restaurant to new user based on the inputs he gives. The user can input his budget or cuisine choice or both. After this user is partitioned using K-means clustering algorithm into a cluster of 30-50 member size ideally. Pearson correlation is used to find similarity of user with the neighbor. The neighbor's weight is multiplied with ratings given by him for a particular item.This gives true state of rating inreference to our user.

2.2.2 Real-Time Collision Risk Estimation based on Pearson's Correlation Coefficient:

A novel approach is presented to PCC based on the PCC variation and by using the temporal coherence between consecutive frames. Here researchers estimate the Collision Risk Estimation (CRE) in dynamic and unknown environments by using a single monocular system.
According to the Pearson's correlation, in a certain analysis window (pair of frames), if the obstacle occupies a big portion of the scene, the PCC threshold tends to be low. Conversely, if the obstacle occupies a small portion of the frame, it means that it is away from the vehicle and the system will have time enough to react.

2.2.3 A note on Pearson Correlation Coefficient as a metric of similarity in recommender system:
Pearson Correlation Coefficient (PCC) is used to evaluate correlation between two users. Earlier Correlation-based prediction schemes performed well but lately some disadvantages are discovered. This paper presents an extension toward Pearson Correlation Coefficient measure for cases which does not exist similarity between users by using it. Experimental result on the film trust data set demonstrate via proposed measure and PCC can achieve better result for similarity measure than traditional PCC.
Using PCC measure, we can filter out some users' pairs that have more or less similarity to rating scores on the same items. Based on the assumption users with similar tastes on different types of products have higher probability to form a community and are likely to make associates to each other even if they don't know each other before.

2.3 GINIX:
2.3.1 Ginix: Generalized Inverted Index for Keyword Search:
Ginix saves storage space by merging consecutive IDs in inverted lists into different intervals. With this index structure, more efficient algorithms can be devised to perform basic keyword search operations, i.e., the union and the intersection operations, by taking the advantage of intervals. These algorithms do not require conversions to ID lists from interval lists. Due to this, keyword search using Ginix is more efficient than traditional inverted indices. Using two scalable algorithms, performance of Ginix is improved. Experiments using the real datasets to assess performance and scalability show that Ginix not only saves storage space but also improves the keyword search performance, compared with traditional inverted indexes.

3. System Architecture Overview

3.1 AES:

There are two types of AES – Symmetric and Asymmetric. Here we will be using Symmetric AES. AES algorithm forms the backbone for encryption in our project. It is found at least six times faster than triple DES.

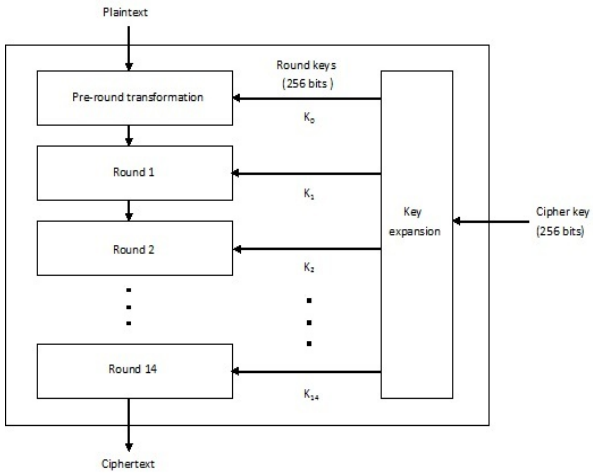

Fig. 1. AES

The Advanced Encryption Standard (AES) also called as FIPS PUB 197 is based on Rijndael cipher. It is a symmetric block cipher used to encrypt and decrypt electronic data. After encryption an unintelligible form of data called ciphertext is generated. Decrypting this ciphertext will revert it back to original plaintext. Different versions of AES are AES-128, AES-192 and AES-256 which use keys of length 128, 192 and 256 bits respectively. All of them encrypt and decrypt blocks of 128 bits of data.

3.2 Pearson Correlation:

Karl Pearson developed Pearson product-moment correlation coefficient sometimes referred to as the PPMCC or PCC or Pearson's *r*. It gives a linear correlation between two variables *X* and *Y*. It gives a value between +1 and −1 inclusive, where 1 is total positive correlation, 0 is no correlation and −1 is total negative correlation. It is widely used as a measure of the degree of linear dependence between two variables.

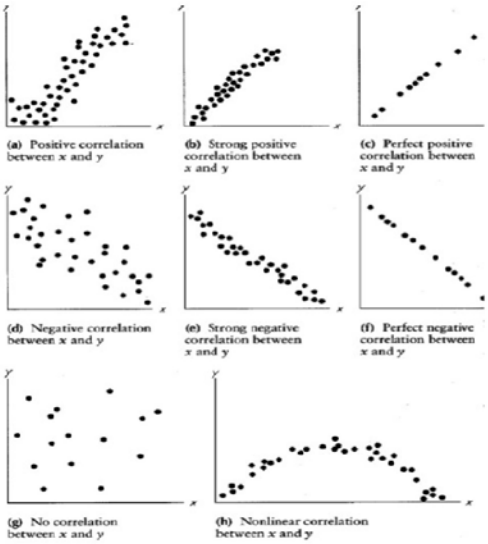

Fig. 2. Pearson Correlation

- **Formula for Pearson Correlation :**

$$r = \frac{\sum_{i=1}^{n} X_i Y_i - \frac{\sum_{i=1}^{n} X_i \sum_{i=1}^{n} Y_i}{n}}{\sqrt{\sum_{i=1}^{n} X_i^2 - \frac{(\sum_{i=1}^{n} X_i)^2}{n}} \sqrt{\sum_{i=1}^{n} Y_i^2 - \frac{(\sum_{i=1}^{n} Y_i)^2}{n}}}$$

Where,

r = Pearson correlation coefficient

n = Total no. of values in each data set

$\sum_{i=1}^{n} X_i Y_i$ = sum of products of paired scores, i = 1 to n

$\sum_{i=1}^{n} X_i$ = sum of x scores

$\sum_{i=1}^{n} Y_i$ = sum of y scores

$\sum_{i=1}^{n} X_i^2$ = sum of squared x scores

$\sum_{i=1}^{n} Y_i^2$ = sum of squared y scores

In our project, Pearson Correlation is used to find similarity between generated trapdoor of query and feature file. We will be providing a threshold value above which the algorithm will consider the results for shortlisting of returned files.

3.3 GINIX:

Efficient retrieval of documents using a set of keywords is based on inverted lists which indexes these underlying documents. The Generalized Inverted Index (Ginix) saves storage space by merging consecutive IDs in inverted lists into different interval. Inverted index stores a set of (key, posting list) pairs. This posting list is a set of documents where key is present.

In this project, Ginix is utilized to minimize the number of results returned after Pearson correlation. The contiguous results are merged and only the boundary objects are returned. For example if files 1-9 and 11,14,19 are returned the the result is displayed as {{1,9},{11},{14},{19}}. In this way the allocated objects are reduced.

4. Methodology :

1. Data Owner: The data owner has a set of sensitive data, he wants to outsource to a cloud server owned by cloud service provider.
2. Cloud Server: Cloud server is the remote service provider, which stores and manages the data generated by data owners. Powerful and intuitive interfaces are given by service provider to data owners and users to create, store, access and manipulate databases. The administration of the database e.g. installation, backups, reorganization, s/w updates .search is done by the service provider.

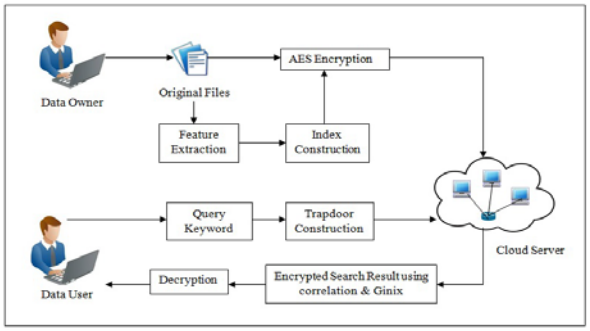

Fig. 3. System Architecture

3. Data User: A data user could either be same as the data owner or if some company or organization is the owner then employees or clients of company could be data users.

The above system consists of following steps:

Step 1: Feature Extraction: Data owner uploads plaintext file. As soon as he uploads file features are extracted from the file by preprocessing it. Stopwords removal and stemming id is done for feature extraction.

Step 2: Index Construction: From the features extracted in the above step index construction is done. In this stage for each word in file buckets are created. Buckets are created by splitting the word at third character and so on till the total length of word. All these words are stored in a file.

Step 3: Encryption: Encryption of both plaintext file and indexed file is done by using advanced encryption standard (AES) algorithm. 256 bit key is used for it. Both encrypted file and encrypted feature file are stored on cloud.
Step 4: Trapdoor Construction: When data user gives query keyword, trapdoor is generated. First query keyword is preprocessed then bucket of that keyword is formed. AES encryption with 256 bit key is applied. Trapdoor helps to search on encrypted database.
Step 5: Search: Cloud server performs search by using Pearson correlation and generalized inverted index. Documents having more similarity to the query keyword are returned back to the user.
Step 6: Decryption: From the files retrieved in search stage user downloads file. User gets the plaintext file after decryption. Advanced encryption standard is used for decryption.

5. RESULTS AND DISCUSSIONS

Some experimental evaluations are performed to show the effectiveness of the system. And these experiments are conducted on windows based java machine with universally used IDE Netbeans. Also the numbers of retrieved documents are used to set benchmark for performance evaluation.
Numbers of relevant retrieved documents from the cloud for the set of keywords are used to show the effectiveness of the system.
Below are the definition of the used measuring techniques i.e. precision and recall.
Precision: it is a ratio of numbers of proper documents retrieved to the sum of total numbers of relevant and irrelevant documents retrieved. Relative effectiveness of the system is well expressed by using precision parameters.
Recall: it is a ratio of total numbers of relevant documents retrieved to the total numbers of relevant documents not retrieved. Absolute accuracy of the system is well narrated by using recall parameter
Numbers of scenarios presents where one measuring parameter dominates the other. by taking such parameters into consideration we used two measuring parameters such as precision and recall.
For more clarity let we assign
• X = the number of relevant documents retrieved,

• Y = the number of relevant documents retrieved are not retrieved, and

• Z = The number of irrelevant documents are retrieved .

So, Precision = (X/ (X+ Z))*100

And Recall = (X/ (X+ Y))*100

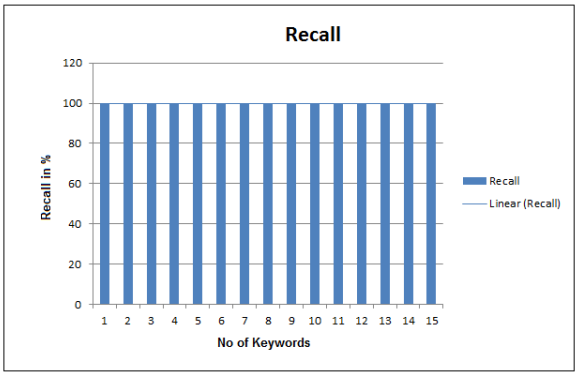

Fig.4. Average precision of the similarity search method

In Fig. 4, by observing figure 4 it is clear that the average precision obtained by using similarity search method is approximately 68%.

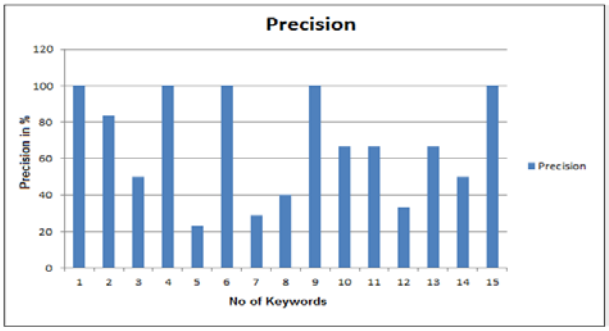

Fig. 5. Average Recall of the similarity search method

In Fig. 5, figure shows that the system gives 100% recall for the similarity search method. By comparing these two graphs we can conclude that the similarity search method gives high recall value compare to the precision value.

6. Conclusion and Future Scope

This project establishes the importance of faster retrieval of data from Cloud. This has become more important in the current scenario where usage of cloud infrastructure is on a rise. As more users move towards cloud for storing their information, it is essential for cloud providers to use newer algorithms which give speedy retrieval without compromising security of user data.

AES provides security to the data stored in cloud. AES is the most trusted algorithm for encryption and we have used it for both data encryption and query encryption. Also Pearson correlation is used to search required query with maximum accuracy. Ginix reduces the space required for storing the required results.

This project provides an easy way for users to upload their data and store it securely in encrypted form.

This project can be extended to implement image storage and retrieval.

7. Acknowledgement

We would like to thank our Guide Prof. U. A. Mande sir for the support and guidance he gave us on every step of the project execution. We would also like to thank the project review committee members Prof. D. D. Gatade mam and Prof. M. P. Wankhede sir who gave us their valuable comments. We would also like to express our gratitude to HOD Prof. P. R. Futane sir who helped us to accomplish this work.

8. Reference

[1] Zhihua Xia, Xinhui Wang, Xingming Sun, Qian Wang, "A Secure and Dynamic Multi-keyword Ranked Search Scheme over Encrypted Cloud Data", IEEE 1045-9219 (c) 2015.

[2] Nasrin Khanezaei, Zurina Mohd Hanapi, "A Framework Based on RSA and AES Encryption Algorithms for Cloud Computing Services", IEEE Conference on Systems, Process and Control (ICSPC 2014), 12 - 14 December 2014.

[3] Anirudh Sunil Nath, Anirudh Sunil Nath, Anirudh Sunil Nath, D. Vanusha, "Implementation of Data Privacy between Nodes Using AES in Wireless Ad Hoc Networks", Proceedings of the second ACM MobiHoc workshop on Airborne networks and communications, pages 19-24, 2013.

[4] Prashant Rewagad, Yogita Pawar, "Use of Digital Signature with Diffie Hellman Key Exchange and AES Encryption Algorithm to Enhance Data Security in Cloud Computing", International Conference on Communication Systems and Network Technologies, 2013.

[5] Yunzhou Zhang, Huiyu Liu, Wenyan Fu, Aichun Zhou, Liang Mi, "Localization Algorithm for GS M Mobiles Based on RSSI and Pearson's Correlation Coefficient", IEEE International Conference on Consumer Electronics (ICCE), 2014.

[6] Hao Wu, Guoliang Li, Lizhu Zhou, "Ginix: Generalized Inverted Index for Keyword Search", TSINGHUA SCIENCE AND TECHNOLOGY ISSN 1007-0214 10/12 pp77-87 Volume 18, Number 1, February 2013.

[7] Qiang Tang, "Search in Encrypted Data: Theoretical Models and Practical Applications", APSIA group, SnT, University of Luxembourg 6, rue Richard Coudenhove-Kalergi, L-1359 Luxembourg, November 14, 2012.

[8] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Enabling Efficient Fuzzy Keyword Search over Encrypted Data in Cloud Computing", in Cryptology ePrint Archive, Report 2009/593, 2009.

[9] H. Park, B. Kim, D. H. Lee, Y. Chung, and J. Zhan, "Secure Similarity Search", in Cryptology ePrint Archive, Report 2007/312, 2007.

[10] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data", in Security and Privacy, 2000. SP 2000. Proceedings. 2000 IEEE Symposium on. IEEE, 2000, pp. 4455.