Secure Multi-Keyword Search Mechanism over Encrypted Cloud Data using AES and Bloom Filter

A M Aniket [1], Amanpreet Singh [2], Shubham Kumar [3], Preetam Kumar [4]

Computer Engineering, Vishwakarma Institute of Information Technology, Savitribai Phule Pune University
Pune-48, Maharashtra, India

aniket.best321@gmail.com, shubhamkumar567@gmail.com

Abstract- Cloud computing has been visualized as the next generation platform for the IT enterprise helping more and more cloud owners to outsource their data to cloud servers. The sensitive data should be encrypted prior to its upload and eventually needs to be decrypted for use leading to higher processing time in cloud service. Therefore, there arises a need to devise a system that reduces searching or retrieval time of files of this cloud data. The existing encryption systems available for this purpose are UDMRS (based on the greedy depth first search), BDMRS (based on the KNN algorithm) and EDMRS. These systems are less cost effective and are low in performance efficiency. Moreover, they encounter frequent encryption and decryption of files thereby, increasing search time. Our proposed methodology will help in reducing the search time over encrypted data drastically by extracting features of a file prior to its upload and doing correlational search over the data, thereby eliminating the process of decrypting a file during the search process. This paper introduces an idea of searching over encrypted data by using correlation and the Bloom filter mechanism.

Index Terms- Inverted Index, Pearson Correlation, AES, Bloom Filter

1. INTRODUCTION

The paper proposes a retrieval system which will reduce the processing time in retrieving files from the cloud by a considerable amount. The system will extract all the features of a file prior to its upload on the cloud server by applying various pre-processing steps. These features and incoming search query are encrypted using a symmetric encryption scheme AES and then the system attempts to match them using Pearson correlation scheme. The relevant matched files are displayed to the user. The user inputs a query, which will be encrypted, to search for the files present on the cloud. Through the Bloom Filter Mechanism, the files will be retrieved in minimum possible time. The existing methodologies are UDMRS and EDMRS which are currently being used for the encryption purpose in the cloud [1]. While there has been considerable research done with respect to this area, the existing techniques consume more time than what is required in obtaining files from the cloud through user queries. To fill this gap, we propose an efficient Bloom Filter data structure based system that considerably reduces search time over encrypted data.

The AES Encryption used in the system, is an important mechanism to encrypt data. It is based on the substitution-permutation network, which is a combination of both the software and hardware. The Rijndael's Encryption Scheme is the source of derivation for the AES, although it has proved to much more advantageous. A fixed block size of 128 bits is incorporated in it, while the key size varies from 128, 192, or 256 bits. The number of repetitions of transformation rounds that convert the input, called the plaintext, into the final output, called the cipher text is dependent on the key size. Variants include, 10 cycles, 12 cycles and 14 cycles for different sizes of bits. Each round consists of several processing steps, each containing four similar but different stages, while one of the stages is dependent on the encryption key itself. To transform cipher text back into the original plain text using the same encryption key, a set of reverse rounds are applied.

Pearson's correlation is a fairly advanced correlation method. In recent years, it has been preferred to other correlation algorithms such as Spearman's [7] and Kendall. It is a product-moment correlation coefficient where two variables X and Y are considered, giving a value between +1 and −1 inclusive, where 1 is total positive correlation, 0 is no correlation, and −1 is total negative correlation.

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[n\Sigma x^2 - (\Sigma x)^2][n\Sigma y^2 - (\Sigma y)^2]}}$$

Eq 1: Pearson's Correlation Expression

Bloom filters have a strong space advantage over other data structures for representing sets, such as self-balancing binary search trees, hash tables, or simple arrays or linked lists of the entries. It is a space-efficient based data structure that is probabilistic in nature. Initially, this technique was used when the amount of data to be used was impractically large.

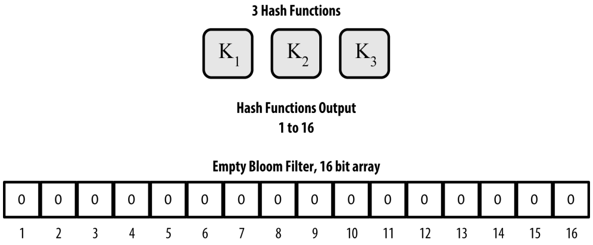

Fig 1: An example of a simplistic bloom filter, with a 16-bit field and three hash functions

The algorithm is extremely unique since it first considers an empty array of m bits. 'k' different hash functions are defined that is used for mapping purposes from the elements in the set to the array. While querying for an element, the element is fed to each of the k hash functions to get k array positions. The element is definitely not in the set of any of the bits are 0, while if all the bits are 1, then the element maybe present in the set.

The rest of the paper can be classified as follows. Section II is dedicated for the Literature Survey while section III describes our proposed technique while section IV depicts the system architecture and its flow description. Finally this paper is concluded in section V.

2. LITERATURE SURVEY

Song et.al. [2] Firstly gave the practical solution for the Searchable encryption on data. They used word by word document encryption. Each word in a document is independently encrypted by using a special two-layered encryption. In this server is given the capability to search on encrypted document and show if the keyword is there in the document. For both keyword (a search with encrypted index) and non-keyword (sequential scan without index) based schemes they gives solutions theoretically. They focus more on keyword based search.

Swaminathan A et.al [3] initiated the use of secure rank-ordered search. For each document in set they collected, term frequency information for building indices. To protect against statistical attacks they secured indices. Depending on encrypted queries it ranks the documents and document having higher ranks will be pushed up using ranked method. It calculates the relevance score for every document and identifies documents which relevant most. They compare the performance of system for search accuracy with system for non-encrypted data. The given method is well suited for large documents and also it provides higher accuracy and security. W Lou et.al [4] address, the issues related to user searches over encrypted data. While traditional techniques support only exact keyword searches, disregarding the importance of mistakes or minor spelling mistakes, fuzzy keyword search enhance the searching process by rendering quickness in search with respect to queries exactly matching predefined inputs or being closest in semantic to these keywords.

N Cao et.al [5] proposed secure and privacy preserving keyword search (SPKS). Their solution is practical and efficient. In this scheme without leaking any information about plaintext, the computational overhead on users is reduced by participating cloud service providers in partial decipherment. They showed that their scheme performs better than the scheme by Bone et al. (2004). This scheme is provably secure and without being aware of any keyword and email information it enables cloud service provider (CSP) for the determination of whether the keyword is present in given email. The main difference between privacy preserving keyword search (PEKS) and SPKS is that in PEKS standard public key encryption algorithm is used to encrypt an email without specifying its specific implementation, and all the decryption is done by a user, whereas SPKS uses the EMBEnc algorithm for encryption of email. It uses a user's public key and the CSP's public key so that CSP can participate in the partial decipherment.

[6] Introduces the use of the Pearson Correlation Coefficient as a means of reducing noise during speech processing. While the Mean Square Error Criterion (MSE) is a fairly unique method to identify noise, the Signal to Noise Ratio behaviour which is an important aspect of noise reduction cannot be identified by the MSE, thereby making the use of the Pearson Correlation Coefficient highly imperative. Analysis of noise reduction performance is comparatively, more favourably accomplished by using the Pearson Correlation coefficient. Pearson Correlation Coefficient (PCC) utilizes the normality of

variables that are analyzed [7]. The quality or how accurately the variables can be related to each other defines PCC. It makes use of quantitative variables. Pearson's Correlation Coefficient assumes that the variables will always have a linear relationship which have to be measured on normal scales. [7] Is based on the significance of the benefits of using Spearman's Correlation Coefficient as compared to PCC and though it is more difficult to implement, the advantages that they present cannot be neglected. The relation between Pearson's Correlation Coefficient and Salton's cosine measure [8] is factored on numerous divisions of the norm of a vector. A threshold value can be set for the cosine by devising an algorithm above which none of the corresponding Pearson correlations would be of a negative value.

Bloom filter has been used as an important method to solve many internet problems [9]. Though they are effective, they are prone to vulnerabilities. Encoding is done via bloom filter which involves source routing of the protocols.

Markku Antikainen et.al. substantiate their opinions on this and present relevant design options for Bloom filter forwarding. [10] Elaborates on the use of Counting Bloom Filters and introduces a method whereby the efficiency of Counting Bloom Filters (CBF) can be improved. Though they consume significant amount of memory, they are used as widely as Bloom filters due to their fast set representations resulting in a less amount of errors, thereby supporting membership queries and element deletions. The increase in the scale of genomic information [11] has been a consequence of rapid advances in next generation sequence technology. Using an Error Correction technique based on Bloom Filter, a new method is introduced which provides accurate result and in comparison with other solutions, uses less memory. While Bloom filter are widely used powerful tools that can be enhanced in the use of processing set member queries, they have certain limitations [12] when it comes to deleting a specific attribute value that may arise when correlation is used in combination, to compare attributes. [12] Proposes a new Bloom filter data structure, based on improved association which maintains the association information on the two correlated attributes of items in the given data set and also makes use of a new hardware coprocessor for a critical part of the algorithm.

AES Encryption has been used as an effective method to encrypt data. It presents various advantages compared to other encryption techniques such as Reverse Cycle Cipher whose execution time derails the mechanism. [13] Introduces a further improved version of the AES Encryption in combination with neural net. Modification of the AES to improve its performance is done with the help of neural network tool that could prove to be fairly advantageous. Ashwini R. Tonde et.al [14] formulate the use of the AES Encryption technique on the basis of an FPGA. The AES is a 128 bit based Encryption technique and the design of this system has been programmed by Very High Speed Integrated Circuits. The results are compared with previous designs and an improved efficiency is shown. Various implementations of AES is possible when testing its use on a multi core system [15]. 16 such implementations of the AES are mapped by exploring different levels of parallelism. The system contrives a design that produces a considerably larger amount of throughput per unit of chip area. It brings about changes in efficiency of the AES when compared with the original standard.

3. PROPOSED SYSTEM

The cloud storage systems are most vulnerable for data security due to their internal data sharing among the servers. To overcome this data is always stored in the cloud by applying strong cryptographic techniques. But eventually this doesn't solve the problem of searching on the cloud data, as cloud is known for storing a huge amount of data, therefore performing search on this huge encrypted data in cloud is complex and poses a real challenge. This paper discusses some novel approaches and also put forwards an idea of increasing speed of searching technique using correlation between the data.

To securely search over encrypted data, searchable encryption techniques have been developed in recent years. Searchable encryption schemes usually build up an index for each keyword of interest and associate the index with the files that contain the keyword, but in the current trend of search over encrypted system in a cloud environment, data to be searched on the cloud undergoes frequent encryption and decryption resulting in the search time being vastly increased. Algorithms such as KNN and Greedy Depth First Search, irrespective of their advantages, have a fair few cons with respect to the problem of search time. Our system will eliminate this tedious process of decrypting every file before it's retrieved by the user. The product is not specified or made to target a certain group of users. Our basic goal concerns the improvements related to any type of Cloud Environments.

Our system should necessarily cover the following objectives:

The system should be able to work in real time in a cloud environment, wherein various client systems are connected to a main server.

The algorithms that are to be used should be efficient enough to encrypt the data that is uploaded in the cloud, either from a single client, or simultaneously from multiple clients. In case a server malfunctions, there should be a facility of replication of the data to another standby system that can act as a server.

In comparison with other techniques, the searching time should be reduced as the actual file is not decrypted while the query is submitted by the user.

The user interface should be easy to understand and operate for the layman users as well. Basically, the main output of our entire system will be the retrieval of file from the query entered by the user, moreover, the amount of time taken for search of the file from the cloud will also be displayed on the user interface. We are using the concept of bloom filter in our proposed system which increases the searching speed tremendously in turn reducing the searching time in cloud.

Bloom filter is a probabilistic data structure which tells us that the given query keyword is either definitely not in the set or may be in the set. The base data structure of a bloom filter is a Bit vector. Each empty cell in that table represents a bit, and the number below it its index. To add a word to the Bloom filter, we simply hash it a few times and set the bits in the bit vector at the index of those hashes to 1. When a query keyword is fired by the user we simply hash the string with the same hash functions see if those values are set in the bit vector. If those bits are not set we can definitely say that elements is not in the set. There are certain drawbacks of the bloom filter in the form of false positives which report that an element is present in the bloom filter even if it is not present in it. We are overcoming this drawback in our proposed system by increasing the number of hash functions and also increasing the size of the bit vector. This will drastically reduce the number of the false positives in our system.

4. SYSTEM ARCHITECTURE

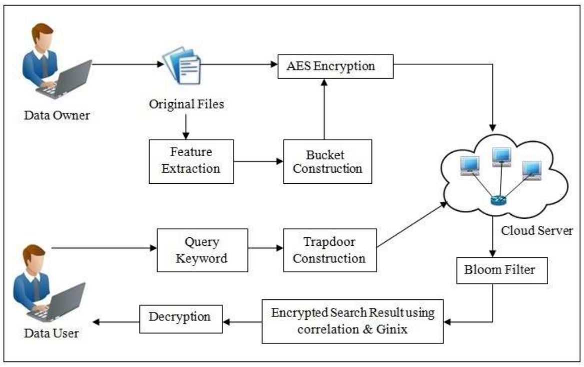

Fig 2: System Architecture/ Data Flow Architecture

The above diagram represents the flow of our system. In our proposed system the client first uploads the file on cloud. The file is processed using various data pre-processing techniques like stop word removal, stemming etc. The features are extracted in this process. The pre-processing is done so as to filter out the common words which are present in almost all the files. It helps in extracting the root word features which is mostly uniquely associated with every file.

A bucket is created which stores all the possible combinations of the contents of the feature extracted file. All possible combinations of the words present in the file after pre-processing is stored in the bucket associated with that word. This helps in those cases when the user has typed the query incorrectly or when the user is not sure of the exact query keywords. After this both the original file and the corresponding feature extracted file are encrypted using the AES Encryption algorithm. The key used in the AES encryption scheme is symmetric in our proposed system.

These files are then uploaded in the cloud where it is replicated across the machines. This is done for fault tolerance as when any of the node in the cloud network is down the client's file can be accessed from the other machines which have the copy of the client's file.

The next phase deals with the searching part of our proposed system. The client queries for a particular keyword term. This query is encrypted using the AES encryption and then the system searches for that query in the bloom filter using Pearson correlation. Trapdoor helps to search on encrypted database. It enables computing in one direction making it difficult to find its inverse or compute in some other direction as opposed to the original one. It's a cryptographic measure. Hashing functions are applied on the query. The index values generated are then checked in the bloom filter. If bloom filter reports that all the corresponding index bits are set in it then our system can derive that the query terms have been found in the cloud. The corresponding files containing the query term are thus found.

The relevant matching files with high degree of similarity are then displayed to the client after. The user can then download the original files once they are decrypted

5. CONCLUSION

With the advent of cloud computing there is a large amount of data to be dealt with. Ultimately most of these files have to be retrieved by the user at some point during certain activity. In this paper a secure, efficient and dynamic search scheme is proposed which will drastically reduce the search time for the retrieval of cloud data furthermore, readily supporting the accurate multi-keyword search mechanism. Moreover the security of the system is ensured since the original file which is uploaded on the cloud is never decrypted.

ACKNOWLEDGMENT

We wish to express our profound and sincere gratitude to Prof. Sachin Sakhare, Head of Department, Department of Computer Engineering, VIIT for his valuable support, guidance and his encouragement in the progress and successful study of this paper. We are also highly grateful to our parents, teachers and all our friends who showed faith in us and supported our endeavor.

REFERENCES

[1] Zhihua Xia, Xinhui Wang, Xingming Sun, and Qian Wang, "A Secure and Dynamic Multi-keyword Ranked Search Scheme over Encrypted Cloud Data", IEEE Transactions on Parallel and Distributed Systems, (Volume: PP, Issue: 99), 11 February 2015

[2] D. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Proc. of the IEEE Symposium on Security and Privacy'00, 2000, 44–55.

[3] Ashwin Swaminathan, Yinian Mao, Guan-Ming Su, Hongmei Gou, Avinash L. Varna, Shan He, Min Wu, Douglas W. Oard, "Confidentiality-preserving rank-ordered search ", Proceedings of the 2007 ACM workshop on Storage security and survivability, Pages 7-12

[4] 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.

[5] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure ranked keyword search over encrypted cloud data," in Proc. of ICDCS'10, 2010, pp.253– 262.

[6] Jacob Benesty, Jingdong Chen, and Yiteng (Arden) Huang, "On the Importance of the Pearson Correlation Coefficient in Noise Reduction", IEEE Transactions on Audio, Speech, and Language processing, (Vol. 16, No. 4), May 2008

[7] Jan Hauke and Tomasz Kossowski," Comparison of Values of Pearson's And Spearman's Correlation Coefficients on the Same Sets of Data", Quaestiones Geographicae 30(2) 2011

[8] Leo Egghe and Loet Leydesdorff,"The relation between Pearson's correlation coefficient and Salton's cosine measure", Journal of the American Society for Information Science & Technology

[9] Markku Antikainen, Tuomas Aura, and Mikko Särelä, "Denial-of-service attacks in bloom-filter-based forwarding", IEEE/ACM Transactions on Networking (TON) archive, Volume 22 Issue 5, October 2014

[10] Ori Rottenstreich, Yossi Kanizo, and Isaac Keslassy, "The variable-increment counting bloom filter", IEEE/ACM Transactions on Networking (TON) archive, Volume 22 Issue 4, August 2014

[11] Yun Heo, Xiao-Long Wu, Deming Chen1, Jian Ma, and Wen-Mei Hwu1, "BLESS: Bloom filter-based error correction solution for high-throughput sequencing reads", Bioinformatics(2014), January 21, 2014

[12] Jiangbo Qian, Qiang Zhu, and Yongli Wang, "Bloom Filter Based Associative Deletion", Parallel IEEE Transactions on Parallel and Distributed Systems, Volume 25, Issue 8, August 2014

[13] Seema Rani, Dr Harish Mittal, "A Compound Algorithm Using Neural and AES for Encryption and Compare it with RSA and existing AES", Journal of Network Communications and Emerging Technologies (JNCET), Volume 3, Issue 1, July (2015)

[14] Ashwini R. Tonde, and Akshay P. Dhande, "Implementation of Advanced Encryption Standard (AES) Algorithm Based on FPGA", International Journal of Current Engineering and Technology E-ISSN 2277 – 4106, P-ISSN 2347 – 5161, 2014

[15] Bin Liu, Baas, B.M, "Parallel AES Encryption Engines for Many-Core Processor Arrays", IEEE Transactions on Computers (Volume: 62, Issue: 3), March 2013