IJITCE

ISSN 2347- 3657

International Journal of Information Technology & Computer Engineering

Email : ijitce.editor@gmail.com or editor@ijitce.com

International Journal of Information Technology & Computer Engineering

ISN NO: 2347 – 3657

Volume 10, Issue 2, 2022

Attribute Based Top K Data Retrieval over Documents in Cloud Storage

Wan Zhe, Lary J. Proko, , Matth C. Henande,

Evidence-based Practice Center, Mayo Clinic, Rochester
Library Public Services, Mayo Clinic, Rochester 5
, Division of Trauma, Critical Care, Department of Surgery, Mayo Clinic, Rochester,

Abstract

One of t he biggest technological developments of our day, cloud computing, provides a wide range of services to meet the demands of numerous users. Service providers charge based on the amount of time spent using their services. Cloud computing becomes more difficult as security and scalability requirements are imposed on cloud data. In spite of its many benefits, cloud computing still has a constraint known as security that has to be solved. Encrypting and uploading sensitive data to the cloud is necessary to ensure the privacy and efficiency of data storage. Searching for files using numerous keywords is necessary when the data gets more complicated and the number of users grows. Traditional searchable encryption techniques enable users to search encrypted cloud data using keywords, which merely permit search, i.e. if a term is in a file or not, without any relation to data files and the searched keyword.. As a consequence of a single term ranked search, data privacy is compromised while searching the cloud utilising server side ranking based on order preservation encryption. The TF/IDF Algorithm was suggested in this Project to find the most relevant K documents.

Keywords: Document retrieval, file hierarchy, and attribute-based encryption are all examples of cloud computing.

1. Introduction

As the information system (IT) grows, more people and businesses are embracing cloud services, which promise a promising data system (IT) to process the ever-expanding amount of information. However, releasing sensitive data, such as personal information and financial information, to the general population is a major risk to the information owner. In order to get the most out of the cloud's data, customers need to be able to access it quickly and easily. Scrambling recordings first and then uploading them to the cloud is an intuitive approach.

Single keyword Boolean search strategies, single keyword ranked search methods and multi-keyword Boolean search schemes have been presented in the literatures. Although each document in these systems is grouped by a coarse-grained access control mechanism, each

ISN NO: 2347 – 3657

Volume 10, Issue 2, 2022

authorised data user may access all of the encrypted documents. The fact that all approved institutions (such as universities) may now access the whole IEEE Explore Digital Library, for example, will not satisfy data owners and consumers in the future. A fresh scenario is examined in this study. Each document in the collection can only be viewed by a limited number of data users.

It is more rational than the current way to devise a fine-grained document access control system. Using attribute-based encryption (ABE) techniques to encrypt documents before storing them in the cloud might be one way to enable data consumers able to access portions of IEEE Explore Digital Library on demand. In the meanwhile, a set of characteristics is allocated to those who have been granted access to the data. Data users can only decode encrypted files if their characteristics match the attributes of the file they're trying to decrypt. Research in recent years has focused on ciphertext policy attributes, which may enable fine-grained, one-to-many, and flexible access control for ciphertexts (CP-ABE). Hierarchical attribute-based encryption techniques increase the encryption efficiency by encrypting each document separately. The difficulty is that we can't use these methods to fix it. Attribute-based access control mechanisms can't currently be supported by the majority of existing schemes, as far as we're aware. In order to enable the previously stated service, we first create an algorithm that generates hierarchical access trees for the document collection Encryption and decryption time and cypher text storage space are both reduced in this manner. The suggested scheme's security and efficacy are both shown theoretically and tested via simulation. A complex index structure is then built for the document collection in order to provide accurate and efficient document search across encrypted documents. To begin, we use the TF-IDF model to convert the documents into document vectors, and then we take into account their properties. The ARF vectors of the tree nodes are utilised to characterise the clusters represented by the nodes. Finally, the ARF tree has a depth-first search algorithm that ensures both efficiency and accuracy in search.

The DES Encryption Algorithm

There are sixteen steps or rounds in the transmission of a message block. The input key is used to produce sixteen 48-bit keys, one for each round. Eight so-called S-boxes are utilised in each round. The standard specifies that certain S-boxes must be used. It is possible to map groups of six bits using S-boxes. According to the US National Security Agency, the contents of these S-boxes have been determined (NSA). Despite the fact that the S-boxes seem to be filled at random, this is not true. These S-boxes, developed in the 1970s, have recently been shown to be immune against a kind of attack known as differential cryptanalysis, which was first identified in the 1990s. The message is split into two halves. Another fixed table is used to enlarge the right half from 32 to 48 bits. Subkeys are joined with the result of that round's XOR operation. It is then turned into 32 bits using S-boxes before being permuted again using a different fixed table. The XOR function is used to merge the right and left halves of

ISN NO: 2347 – 3657

Volume 10, Issue 2, 2022

this now fully scrambled data set. This combination is utilised as the new left half in the next tournament.

2. **DES Algorithm**

1a. 64-bit key and 64-bit plain text input to generate 64-bit ciphertext. Data bits of left 32 bits and right 32 bits are distributed equally between the two halves.

Every 8th bit in the 64-bit system is used for parity testing. So 56 bits are taken into account.

64 bit input is delivered to the first permutation, which generates 64 bit data. Using permutation choice one (PC-1), the 64-bit key is reduced to 56 bits, with multiples of 8 being disregarded.

In this case, the 56-bit key is split into two equal 28-bit keys.

2a. The PC-2 generates a 48-bit round key using these 56 bits. The expansion permutation takes a 32-bit round key and turns it into a 48-bit round key.

To produce a 48-bit block, the PC-2 output is XORed with an expansion permutation.

2c. These 48 bits are separated into eight portions and transmitted to the substitution blocks (S-BOX). A total of 32 bits are created by eliminating two bits from every S-BOX.

A permutation function receives these S-BOX bits once again and performs an XOR operation on 32 bits.

Third, XOR output is connected to the Ri and Li of the following round, which in turn are linked to the Ri of these rounds. Operations 1, 2,9,16 have just one bit left circular shifting, whereas the next 12 rounds have two-bit left circular shifting. ' Final 32 bit switching and feeding to inverse permutation are conducted after 64 bits have been processed in 16 cycles here. The file is encrypted using the DES Encryption Algorithm. We've also incorporated stop words, stemming, and the TF/IDF technique in order to increase the search's performance.

Stop words

Stop words are used to eliminate unnecessary information from a document. The target document's text is tokenized, and each word is saved in a separate array. A single word from the stop word list is read aloud. Using a sequential search strategy, the stop word is compared to the target text in the form of an array. The word in the array is eliminated if the two are same, and the comparison is carried on until the array reaches its maximum length.

Stemming

When words are inflected (or sometimes derived), they are reduced to their word stem, base, or root form. A word's morphological root need not be the same as the word's stem; in most cases, it suffices if related words all map to the same stem.

As an example, computing is transformed into computation.

ISN NO: 2347 – 3657

Volume 10, Issue 2, 2022

TF-IDF (Relevance Count)

When it comes to information retrieval in the form of tf–idf, short for term frequency–inverse document frequency, tf–idf is an indicator of the importance of words in the collection or corpus. The tf-idf word weighting system has become more popular in recent years. The tf–idf weighting technique is often used by search engines to score and rank the relevance of a page to a user query. A variety of domains may benefit from Tf–idf stop-words filtering, such as summarization and classification. Summing the tf–idf for each query phrase is one way to construct a basic ranking function; several more complex ranking functions are based on this basic concept.

3. Conclusion

Encrypted document retrieval scenarios in which the data owner desires fine-grained management of the documents are discussed in this work. A unique hierarchical attribute-based document encryption system is designed to encrypt a collection of documents with an integrated access structure in order to enable this service. To further arrange document vectors, the ARF tree has been suggested.

It is now possible to increase data consumers' search efficiency by using a depth-first search algorithm for huge document collections. Experiments and theoretical analysis are used to assess the approach's effectiveness.

The CA centre and the cloud server are taken for granted in this project. Data users will get accurate characteristics from CA centres, and cloud servers will carry out all commands honestly. Data consumers are also assumed to be hungry and strive to collect as many unencrypted files they can. The encrypted papers are being decrypted with the help of other users. We focus primarily on the encryption, document search, and decryption processes.

References

[1] In "Security issues for the public cloud," by K. Ren, C. Wang, and Q. Wang, IEEE Internet Computing, January 2011, pages 69–73. (2012).

[2] "Practical strategies for searching encrypted data" was published in the 2000 issue of "Security and Privacy" by D. X. Song, D. Wagner and A. Perrig. Pages 0–44 in SandP 2000 Proceedings: Proceedings of the IEEE Symposium on (2002).

[3] A secure index was developed by E J Goh, "Cryptology EPrint Archive," http://eprint.iacr.org/2003/216 (2003).

[4] Searchable symmetric encryption: better concepts and efficient implementations" was presented at the ACM Conference on Computer and Communications Security (ACM CCS), pages 79–88, in 2010. (2006).

[5] Searchable cypher text-policy attribute-based encryption with revocation in cloud storage by J. Li, Y. Shi and Y. Zhang, International Journal of Communication Systems, vol. 30, no. 1, pp. 1-3, (2017).

ISN NO: 2347 – 3657

Volume 10, Issue 2, 2022

[6] Six researchers published an article entitled "Attribute-based keyword search across hierarchical data in cloud computing," published in the IEEE Transactions on Services Computing (PP, no. 99), in which they describe their approach to attribute-based keyword search (2017).

[7]Swaminathan, Y. Mao, G. M. Su, H. Gou, A. L. Varna, S. He, M. Wu and D W Oard, "Confidentiality-preserving rank-ordered search," in Storagess 2007, Alexandria Va, United States, October, pp. 7–12. (2007).

[8] Enabling safe and fast ranking keyword search across outsourced cloud data" was published by Wang, N. Cao, K. Ren, and W. Lou in the August 2014 issue of IEEE Transactions on Parallel and Distributed Systems (TPDS) (2012).

[9] Zerr +r:topk retrieval from a secret index (2006).

[10] Secure conjunctive keyword search over encrypted data, P. Golle, J. Staddon, and B. Waters, Lecture Notes in Computer Science, vol. 3089, pp. 31–45 (2001). (2004).