International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 05 | May-2016 www.irjet.net p-ISSN: 2395-0072

A VIRTUAL KEY STEP TO BOOST AND ENHANCE QUALITY OF SERVICES IN CLOUD COMPUTING

Manvinder Singh[1], M.Nithiya[2], M.Saravanan[3], S.Karthik[4]

[1]*PG Student, Computer Science and Engineering, VMKV Engineering College, Salem, India*
[2, 3, 4] *Associate professor, Computer Science and Engineering, VMKV Engineering College, Salem, India*

Abstract - *In today's world cloud services are growing rapidly. The growing nature of cloud computing made it more popular. Cloud computing as an important functionality called Data Sharing. To share and to search data among various user different encryption keys are used. Users have to secure the received keys to perform share or search operation. Larger number of encryption keys create an equal amount of keyword trapdoor. This process is complex and consume a huge amount of time. In our paper we have tried to rectify the persisting problem through the concept of Key Aggregate Searchable Encryption (KASE). In KASE scheme a data owner will provide a single virtual key to the user to search and share a number of documents. The user's needs to submit a single key in order to generate a single keyword trapdoor. Through this single virtual key user can decrypt a set of the document and other the document will remain confidential and secure. We have also used the concept of blowfish algorithm which is an alternative for aging DES. Our proposed model help in reducing overall operation delay and enhance Quality of Service (QoS).*

Key Words: **Encrypted content Search, Trapdoor, Cloud Storage, Single Virtual key.**

1. INTRODUCTION

In a cloud computing shared resources, data and information are made available for other user's on-demand. The On-demand characteristic of Cloud Computing made it more favorable [10]. Today millions of people can share their personal document, photos or videos with other through cloud storage. But sharing of data in a cloud environment can cause data revealing. To provide security against data revealing data are store in cloud storage in an encrypted form (to provide security). Only specific user those whose have a decryption key can decrypt the document. This type of cloud storage is known as Cryptographic Cloud Storage [9].

Encrypted data are hard to search. To solve such a problem Searchable Symmetric Scheme [1] are deployed. In this scheme data owner will gather some important keyword, decrypt the keyword and upload the same in the cloud storage with other data which make the search much easier.

Cloud computing holds a vital feature of Data Sharing. Data Sharing [6] is the ability to share the same data or resources with multiple users. Data Sharing is a complex and time-consuming propose. First of all data are encrypted and store in cloud storage. Sharing data with multiple users require different encryption key for different data files. Data Owner will provide different keys to the user, to search and to decrypt different files. A large number of key will create an equal amount of keyword trapdoors. User can view and decrypt on file at a time. This process is complex .

In this paper we will introduce a single virtual key concept. In this concept, the data owner will provide a single key to the users to share a huge number of data files. The user needs to generate a single trapdoor for querying the shared data. We have also used blowfish encryption algorithm to fasten the data encryption process. The search time and network evaluation performance test and result confirm that our scheme is more secure and efficient.

2. RELATED WORK

To provide data security and to sharpen search efficiency various research have been conducted in recent years. Yang et al. [2] Propose a model in which multiple users can search and encrypt data, and defines the security requirement. Under their model, each user will have distinct keys provided by data owner for designing search queries. No one else other than the authorized user can search the query. Their main focus is to provide a secure transaction in the cloud environment. In our work apart from providing security we have also concentrated on the quality of the service.

Wang et al. [11] propose the concept of ranked keyword search over encrypted files. According to them, ranked keyword search files will be preferring to a certain keyword and also the matching files. Their work mainly target on searching encrypted data in the cloud environment. Our work along with searching of encrypted data also takes into account sharing of data among a group of user's. Lopez et al. [4] designed a Fully Homomorphic Encryption (FHE) scheme in which users with different keys

can calculate a function over encrypted data using different keys. However, the decryption requires all the parties to come together and run an MPC protocol. This process is impractical, inefficient required a more amount of time.

Popa et al. [3] Proposed the concept of Multi-key Searchable encryption. Their work aim at providing the multiple key to the user to search and Encrypt data. Multiple keys generate multiple trapdoor. Chu et al. [5] employed a method to reduce the number of encryption key. To share the data files with the same user requires different encryption key. Data owner need to provide several keys which is inefficient. To overcome such scenario we will implement a single virtual key method in which user can view, search and decrypt multiple file with a single key. Bao et al. [8] proposed model which is much closer our proposed model. In their work they have employed a technique in which a user's will have different keys but all the data encryption is done with one key and the search over encrypted data are done with one key. In our model we have used a single key for both search and encryption.

3. EXISTING ENCRYPTED SEARCH MODEL

3.1 Working Scenario

In Existing Encrypted Search, Data owner will collect multiple data from multiple sources and encrypt the collected data using different encryption key and store it in the cloud storage. To share uploaded document, with the other user's data owner will have to provide different keys to the user's to be used for different documents uploaded in cloud storage.

However, a user have to securely keep the different keys received. With the number of keys received, a user will have to generate an equal number of keyword trapdoor in order to search, view and decrypt the document .

3.2 Issues in Existing Encrypted Search Model

- In an existing system different keys are distributed among users to decrypt and search data in a cloud environment. To keep all received keys securely is a difficult task for a users, and it may also raise the concern of the system security requirement.
- With the large number of received keys a user have to generate an equal number of keyword trapdoors to perform a keyword search and encryption over different files. This is a complex and a time consuming process.

4. ADVANCED ENCRYPTED SEARCH MODEL DESIGN

4.1 System Architecture

In the Advanced Encrypted Search Model we had address the persisting problem with the existing system. In this paper we have proposed a concept of Key Aggregate Searchable Encryption (KASE) [7] to make data sharing more secure and to prevent against data revealing. As shown in Fig-1 a data owner will collect multiple data from multiple source and then decrypt the collected data using different decryption keys. In decryption, plain text is converted into Ciphertext using an algorithm called cipher. Ciphertext is a form of plain text which is unreadable by other users until they have a secret key to decrypt it.

The users will request the data owner to provide the secret key to decrypt the data .The Data Owner will then only needs to provide a single virtual key to a user to share the multiple documents in cloud environment. The single virtual key is an aggregated key which consist of the power of all secret keys.

With the single virtual key provided by the data owner, user needs to generate a single keyword trapdoor to the cloud in order to search and decrypt the data. Using the single virtual key user can then view, decrypt and download the data as per their requirement. Our proposed model is simpler, secure, and efficient and consume less time as compare to the existing encrypted search model.

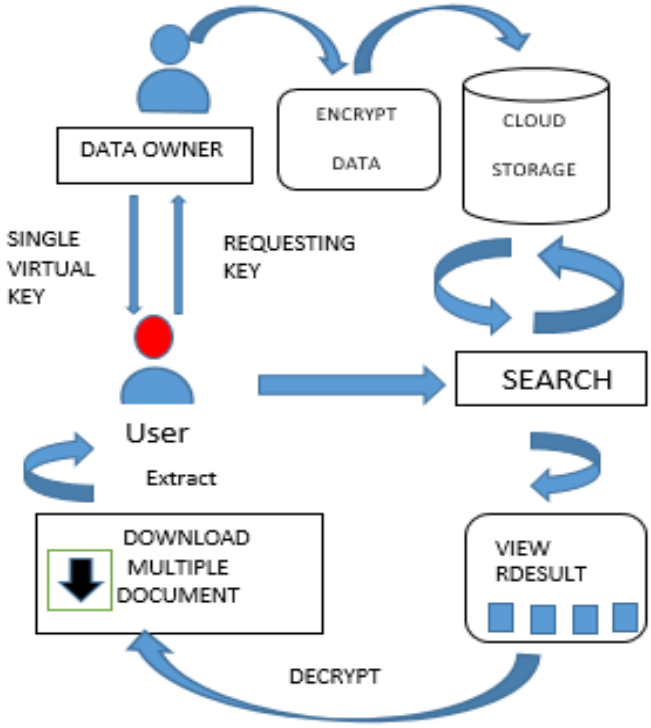

Fig-1: Proposed System Architecture

4.2 KASE Scheme

The KASE scheme consist of seven algorithms.

i. **Setup** (1n, m): The Service provider will schedule and run the set up algorithm. 1n is the security

parameter and m is the maximum number of document belongs to the data owner. 1n and m are the input parameter, its output is a public system parameter p.

ii. **Keygen** (ek, msk): Data owner will schedule and run the keygen algorithm. Here, data owner will generate a pair of keys[ek, msk]. A ek is the encryption key and msk is the master secret key.

iii. **Encrypt** (ek, i): In this, Data owner will encrypt all the document and evaluate their keyword cipertext. On the input of the owner's encryption key ek and the file index i, the output will be data cipertext and keyword cipertexts.

iv. **Extract** (msk, S): The data owner run this algorithm to generate a single virtual key (svk). Input of the algorithm is msk and set S which enclose the directory of the document, and output the svk.

v. **Trapdoor** (svk, k): The user having the single virtual key will run this algorithm. The inputs of the algorithm is svk and a keyword k, it will generate a single trapdoor has an output td.

vi. **Adjust** (p, i, S, td): This algorithm generate appropriate keyword trapdoor for each document. Its inputs are p, i, S and td and outputs each trapdoor [td_i] for the i-th target document in S.

vii. **Test** (td_i, i): Cloud Server will run this algorithm. In this algorithm keyword over encrypted data are performed. Its inputs are td_i and i and the outputs denotes whether the keyword k exist in document doc or not.

4.3 Blowfish Algorithm

Blowfish encryption algorithm plays an important role in securing the data in a cloud environment. Blowfish algorithm [12] was designed by Bruce Scheier in the year 1993. It is an alternative for DES, AES and other existing encryption algorithm. Blowfish algorithm is license-free and is available free for all users. It is the fastest block cipher in public use. It set up an ideal environment for the product to be used in mobile phone, notebook, personal computer etc.

Fig 2. Depict that the blowfish algorithm consist of 64-bit block size and a key length of 32 bits to 448 bits. Blowfish has 16-round Feistel cipher and uses fixed S-boxes.

The diagram to the left depict the action of Blowfish. Each line represents 32 bits. The blowfish algorithm consist of 18-entry P-array and four 256-entry S-boxes. Each S-boxes takes 8-bit input and gives 32-bit output. In every cycle one entry of the P-array is used, and once all the cycle is completed, then each half of the data block is XORed with remaining unused P-entries.

The right side of the diagram shows Blowfish's F-function. This function generate four eight-bit input by dividing the 32-bit input, and provide the same as an input to the S-boxes. Modulo 2^{32} are added to the outputs and then XORed in order to produce the 32-bit output.

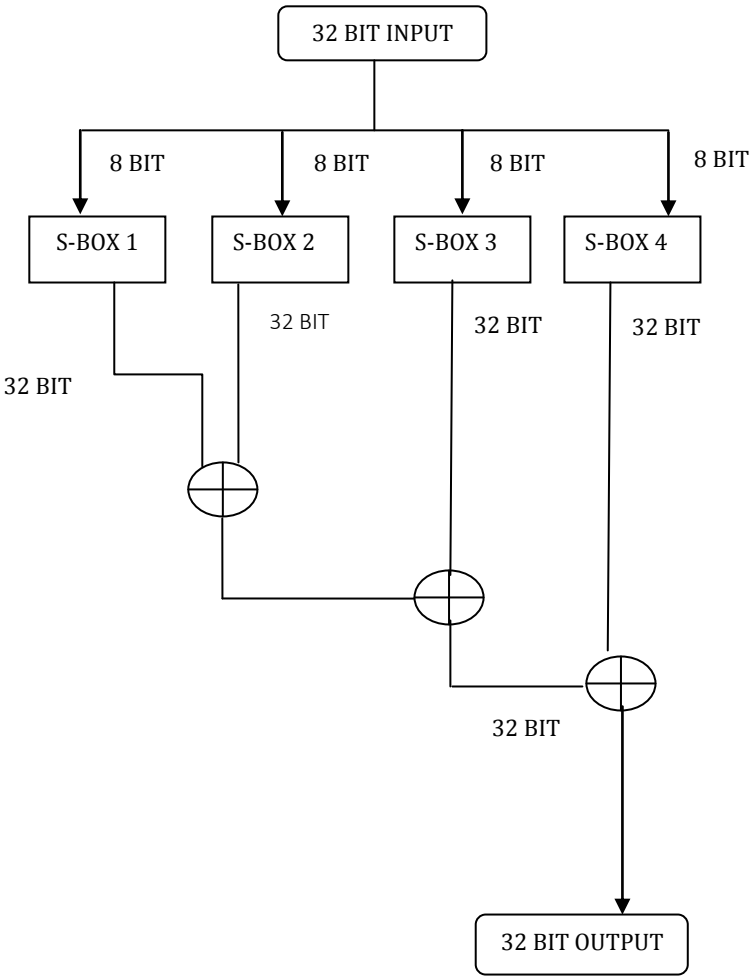

Fig 2- Blowfish Algorithm

Blowfish Algorithm consist of two parts:

- Key expansion, and
- Data Encryption.

4.3.1 Key Expansion

It splits the key into a set of sub keys. It can convert a key of 448 bits into several sub keys totally 4168 bytes. The keys are generated prior to any encryption and decryption. P-array consists of 18 sub keys of 32 bit each. There are four 32 bit S-boxes, which consist of 256 entries in each box.

Algorithm 1: Generating Subkeys

Step 1: Initializing P-array and 4 S-boxes with a fixed string Consists of hexadecimal digit (pi)
Step 2: With first 32- bits of the key P_1 are XOR, then with the Second 32- bits of the key P_2 are XOR and so on (Almost till p14). The step is repeated till all the P- Array has been XORed.
Step 3: All the non-zero string are encrypted, using the sub Key generating in steps (1) and (2).
Step 4: From the output generated in step (3) change P_1 and P_2.
Step 5: Using the sub keys and blowfish algorithm encrypt The output step (3).
Step 6: From the output generated in Step (5) change P_1 and P_2.
Step 7: Process is continue till replacing all the entries in P- Array and then four S-boxes.

4.3.2 Data Encryption

It consist of a function that can iterate 16 times. In each round of iteration key dependent permutation and data dependent substitution are performed. Each and every process are XORs. Other Operations are performed on indexed array of data lookup tables in each round.

Algorithm 2 Data Encryption Algorithm

Step 1: Blowfish consist of 16 rounds. The input of the 64-bit Data element, as n
Step 2: Divide n in two part 32-bit each: nL (left of blowfish), nR (right of blowfish).
Step 3: for i = 1 to 16:
nL = nL XOR Pi
nR = F (nL) XOR nR
Step 4: Interchange nL and nR
Step 5: After all the round, interchange nL and nR to Undo the last interchange.
Step 6: Place nR = nR XOR P17 and nL = nL XOR P18.
Step 7: At the end, combine nL and nR to have the ciphertext.

4.4 System Quality and Overall Delay:

The service quality experience by users is calculated by overall delay cause in searching and decrypting the encrypted data. Larger the amount of decryption key will generate an equally larger amount of trapdoor. A Huge amount of trapdoor causes delay. User will generate each trapdoor in order to view and encrypt a single file. No user will wait for a long period of time if a system cause. In our model through the concept of single virtual key a user can search and encrypt a file by generating a single trapdoor keyword in order to view and encrypt multiple files from cloud storage

5. RESULT AND DISCUSSION

5.1 Experimental Environment

To set up the experiment we have conducted a simple experiment to show the comparison result between our proposed system and existing system. We have used JSP to code our program and MySQL as a cloud storage. To search and retrieved the document we have used KASE method. We have used a PC with an internet speed of 1Mbps rate.

5.2 Search Time Calculation

The KASE scheme is used to reduce the time elapse. It chooses 13000 keywords to search the target document ranging from 1KB to 10KB. Chart-1 shows that our experiment saves 40% of the time as compared to existing method for 1 Kilo Byte document and by 35% for 10 Kilo byte documents. These experiment shows the reduction overall delay.

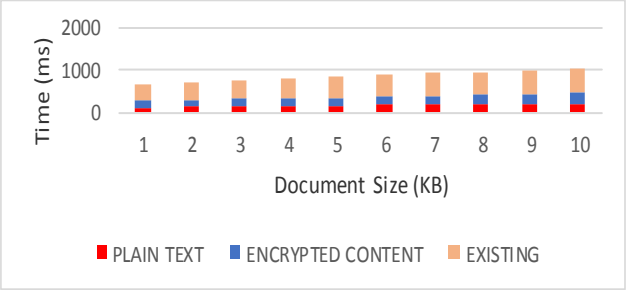

Chart -1: Search Time Evaluation

5.3 Transmission Speed

Larger the amount of trapdoor causes delay. As in our experiment we are using a single virtual key to generate a single trapdoor in order to search and decrypt multiple file. Generating a single trapdoor will boost the overall transmission speed. Chart-2 plain text, encrypted content, and an existing system. The transmission speed for the 1KB-size document is more efficient, and the pace raises from 40 KB/s to 70 KB/s.

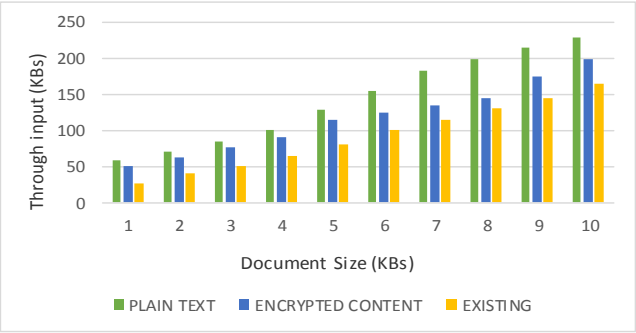

Chart-2: Transmission Speed Result

6. CONCLUSIONS

Taking into account the problem related to Existing Encrypted Search which requires a data owner encrypt and upload the data in the cloud storage and provide a number of keys to users to encrypt the documents , to overcome such situation we propose the concept of key-aggregate searchable encryption (KASE) and Blowfish algorithm. Both confirm that our work can provide an effective solution to the persisting problem. In a KASE method, the owner will encrypt and upload the file in cloud storage. Then data owner distribute a single virtual key to a user to view and encrypt multiple documents by submitting a single trapdoor. Blowfish algorithm provides a fast, alternative to existing encryption algorithms .

REFERENCES

[1] R. Curtmola, J. Garay, S. Kamara, R. Ostrovsky. "Searchable symmetric encryption: improved definitions and efficient constructions" Proceedings of the 13th ACM conference on Computer and communications security, CCS '06, ACM, 2006.

[2] Y.Yang. H.Lu, and J.Weng "Multi-User private keyword search for cloud computing", In IEEE International Conf. on Cloud Computing Technology and Science, pp 264-271.

[3] R,A, Popa, N. Zeldpvich, "Multiple-Key searchable encryption".

[4] A. Lopez-Alt, E.Tromer, V.Vaikuntanathan "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption",In 12 Proceedings of the forty-fourth annual ACM symposium on Theory of computing, pp 1219-1234, 2012.

[5] C. Chu, S. Chow,W. Tzeng, et al. "Key-Aggregate Cryptosystem of for Scalable Data Sharing in Cloud Storage", IEEE Transactions on Parallel and Distributed Systems, vol. 25, issue 2, pp 468-477, 2013.

[6] E.Madhavarao, M.Parimala, C.JayaRaju, "Data sharing in the cloud using distributed accountability", In IJARCET, vol. 2, issue 6, Jun. 2013.

[7] K.Manohar, R. Anil Kumar, Dr. S. Prem Kumar, "Key Aggregate Searchable Encryption for group data sharing via cloud data storage", International Journal of Computer Engineering in Research Trends, vol. 2, issue 12, pp. 1132-1136, 2015.

[8] Feng Bao, Robert H. Deng, Xuhua Ding, and Yanjiang Yang, "Private query on encrypted data in multi-user settings". In ISPEC, pp 71–85, 2008.

[9] S. Kamara and K. Lauter, "Cryptographic cloud storage," in RLCPS, LNCS. Springer, Heidelberg, January 2010.

[10] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds: towards a cloud definition," ACM SIGCOMM Comput. Commun. Rev., vol. 39, pp. 50–55, 2009.

[11] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou. "Secure ranked keyword search over encrypted data",Proc. IEEE International Conf. on Distributed Computing Systems, ICDCS'10,pp. 253-262,2010. pp. 264-271, Dec. 2011.

[12] G.Singh, A. Kumar, K.S.Sandha, " A study of new trends in blowfish algorithm", International Journal of Engineering Research and Applications, vol. 1, Issue 2, pp. 321-326.

BIOGRAPHIES

Manvinder Singh has received his Master of Computer Application from Sikkim Manipal University, India and he is currently studying the Master of Engineering in Computer Science and engineering from Vinayaka Missions University, India.

M.Saravanan received his Ph.D degree in Information and Communication Engineering in the domain Wireless Network Sensor from Anna University, India in 2015. He is working as an Associate Professor Department of Computer Science and Engineering, VMKV Engineering College, Salem, Tamil Nadu.

M.Nithiya has received her Ph.D degree in the area of Grid Computing at Anna University, Tamil Nadu, India in 2012. She is currently Professor in department of Computer Science and Engineering at the VMKV Engineering Colllege, Salem, India

S. Karthik was received his PhD degree in the area of Communication and Networking at Anna University, India in 2015. He is currently an associate professor in the department of Information Technology at the VMKV Engineering College, Salem, India.