PROJECT-1 Report on

**“FUZZY KEYWORD SEARCH OVER ENCRYPTED DATA IN CLOUD”**

Submitted in partial fulfilment of the requirements for the degree of

**BACHELOR OF ENGINEERING**

**in**

**INFORMATION TECHNOLOGY**

**by**

Roll No. Group No: 12

136 Saumya Sharma

137 Parth Agarwal

138 Amrita Bhagtani

139 Ankit Mohite

Under the Guidance of

Ms. Shachi Natu

(Assistant Professor)



Department of Information Technology

Thadomal Shahani Engineering College

University Of Mumbai

2016-2017

**CERTIFICATE**

This is to certify that Saumya Sharma, Parth Agarwal, Amrita Bhagtani and Ankit Mohite with Roll Nos. 136, 137, 138 and 139 respectively has satisfactorily carried out the Project-A work entitled “**FUZZY KEYWORD SEARCH OVER ENCRYPTED DATA IN CLOUD**” as a part of curriculum for the degree of Bachelor of Engineering in Information Technology under University of Mumbai.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Prof. Shachi Natu Dr. Madhuri Rao

Assistant Professor, Head, Associate Professor

Department IT, TSEC Department of IT, TSEC

Project Entitled: **FUZZY KEYWORD SEARCH OVER ENCRYPTED DATA IN CLOUD**

Is submitted in partial fulfilment of the requirements for the degree of

**BACHELOR OF ENGINEERING**

**in**

**INFORMATION TECHNOLOGY**

**by**

Saumya Sharma

Parth Agarwal

Amrita Bhagtani

Ankit Mohite

**Examiners:**

Internal Examiner: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Sign: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

External Examiner:\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Sign: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Table of Contents**

**List of Figures i   
 List of Abbreviations ii   
 Abstract iii**

1. **Introduction** 1
   1. Problem definition 2
   2. Relevance of the project 2
   3. Scope of the project3
2. **Review of Literature** 4
3. **Description** 11
   1. Brief Description 11
   2. Proposed System 12
   3. Methodology and Analysis 14
      1. User Data Upload Scenario 15
      2. User Data Download Scenario 16
   4. Design Approach 16
4. **Implementation** 20
   1. Software Requirement 20
   2. Hardware Requirement 20
5. **Further Work** 21
6. **References** 22
7. **Acknowledgements** 23

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Fig. No** | **Name of Figure** | **Page No.** |
| 1.1 | Architecture of fuzzy keyword search | 2 |
| 2.1 | Previous scenario (Single server) | 6 |
| 2.2 | Current scenario (Two Server) | 7 |
| 3.1 | Architecture of data storage and retrieval | 13 |
| 3.2 | Use case of data storage and retrieval in cloud | 14 |
| 3.3 | (a) List of file name in cloud (b) N(2) gram list of cloud file name(c) N(2) gram list of searched query (d) frequency and score (e) displayed file rank-wise | 18 |

**List of Abbreviations**

AES Advanced Encryption Standard

DH Diffie Helman

SHA Secure Hash Algorithm

**ABSTRACT**

With recent significant development in the portable device market, cloud computing is getting more and more utilized. Many sensitive data are stored in cloud central servers. To ensure privacy, these data are usually encrypted before being uploaded—making file searching complicated. Although previous cloud computing searchable encryption schemes allow users to search encrypted data by keywords securely, these techniques only support exact keyword search and will fail if there are some spelling errors or if some morphological variants of words are used. We provide the solution for fuzzy keyword search over encrypted cloud data. Fuzzy keyword search greatly enhances system usability by returning the matching files when users searching inputs exactly match the predefined keywords or the closest possible matching files based on keyword similarity semantics, when exact match fails. For security reasons, we use two separate servers that cannot communicate with each other. Our experiment result shows that our system is reliable and secure with fast retrieval of data.

**1. Introduction**

Cloud computing, the new term for the long dreamed vision of computing as a utility, enables convenient, on-demand network access to a centralized pool of configurable computing resources (e.g., networks, applications, and services) that can be rapidly deployed with great efficiency and minimal management overhead. As Cloud Computing becomes prevalent, more and more sensitive information are being centralized into the cloud, such as emails, personal health records, private videos and photos, company finance data, government documents, etc. By storing their data into the cloud, the data owners can be relieved from the burden of data storage and maintenance so as to enjoy the on-demand high quality data storage service.

We focus on enabling effective yet privacy-preserving fuzzy keyword search in Cloud Computing. To the best of our knowledge, we formalize for the first time the problem of effective fuzzy keyword search over encrypted cloud data while maintaining keyword privacy.

Fuzzy keyword search greatly enhances system usability by returning the matching files when users searching inputs exactly match the predefined keywords or the closest possible matching files based on keyword similarity semantics, when exact match fails.[1]A gram-based technique, for the construction of fuzzy keyword sets. Both techniques eliminate the need for enumerating all the fuzzy keywords and the resulted size of the fuzzy keyword sets is significantly reduced.

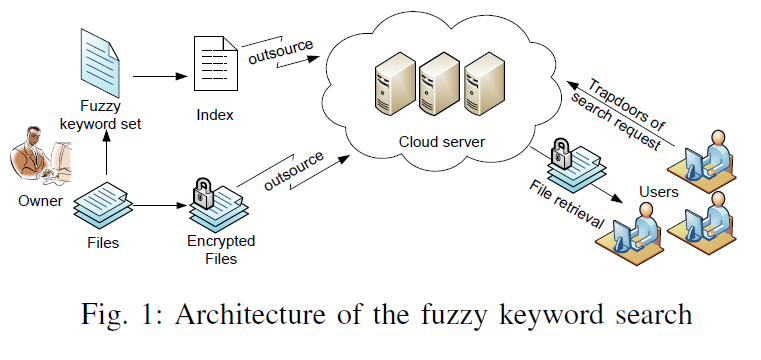
Through rigorous security analysis, we show that the proposed solution is secure and privacy-preserving, while correctly realizing the goal of fuzzy keyword search.

* 1. **Problem Definition**

Motivation for this project is to gain knowledge in the field of cloud computing. Cloud computing is vulnerable in terms of security. Our motive is to encrypt data using public key encryption and further perform fuzzy search. While browsing, users may mistype words due to human error or while typing fast, thus we're performing fuzzy keyword search over the encrypted data. Any user who wishes to upload data securely on cloud, fuzzy search is implemented on this stored data. This product works on the same principle as that of Dropbox but also implements fuzzy keyword search. This product is to establish an effective fuzzy keyword search over encrypted cloud data while maintaining keyword privacy so that unauthorized users cannot gain access to the private files. This product will be more secure and imminent to malware attacks.

* 1. **Relevance of Project**

Over the past few years, many people have started to use cloud computing services for their works. With cloud computing, people can store, access and  
share their information anywhere anytime. Therefore, there are more and more sensitive information being stored in the cloud. In such open environment, users expect not only efficient operations, but also guaranteed privacy and security from the cloud service provider. Our objective is to build a security service which will be provided with a trusted 3rd party, and would lead to providing only security services and wouldn’t store any data in its system.



**Fig. 1.1: Architecture of fuzzy keyword search**

Fig. 1.1 shows the architecture of fuzzy keyword search. In this fuzzy keyword sets of files are made and stored in the cloud server. Fuzzy keyword sets of users search query is made and compared with the sets stored.

* 1. **Scope of the Project**

The future scope of the system is to do the indexing of the mapped words and fuzzy set so as to increase the functionality of the search procedure. Encryption of file formats can be done and it also decryption of images file can also done .

**2. Review of Literature**

In this section we will be talking about the different techniques that previously have been used for data storage like one storage server for storing both key and encrypted data, different fuzzy keyword search technique like wildcard based and N(3) gram based technique. We will be talking about how our new model with 2 server (security and storage server) and N(2) gram fuzzy keyword search is better.

* 1. **Overview of cloud**

In cloud the data is stored centrally, and there are various types of data stored over the cloud such as user’s files, social accounts, game data, website login, and many more. The cloud is used because it helps the data owners a relief from storing of data at their place, because storing the data on our own side may be fatal some times because of hard-disk failure or any other related problems. So for secure storage and retrieval of information such as users files and data we need to encrypt the data before that particular data is stored on the cloud.

Traditional searchable encryption schemes allow a user to securely store data over cloud. Searching over encrypted data through keywords and selectively retrieving files of interest, these techniques support only exact keyword search.

On the other hand users may make a mistake while typing the name of file and this may happen very frequently. This significant drawback makes existing techniques unsuitable in Cloud Computing as it greatly affects system usability, rendering user searching experiences very frustrating and system efficiency very low.

To overcome this we are using fuzzy keyword search technique in our project. As a result even if the user while retrieving the file from cloud makes minor mistake in typing the name of the file then also the desired file is retrieved.

* 1. **Study of security issues in cloud**

Since the administrator has access to data stored in the cloud, they can unintentionally or intentionally access the client data. Security issues which are of concern to the client can be classified into sensitive data access, data segregation, bug exploitation, recovery, accountability, malicious insiders, and account control issues.

Following are some of the concerns:

1. **System Complexity**

Compared to traditional data centre the cloud architecture is much more complex. Therefore while considering security, security of all these components and interaction of these components with each other needs to be addressed.

**2. Shared Multi-tenant Environment**

Since the cloud need to provide service to millions of client, a logical separation of data is done at different level of the application stack . Because of which a attacker in the face off client can exploit the bugs gaining access to data from other organizations.

**3. Internet-facing Services**

The cloud service which is accessed over the internet via browser, the quality of service delivered on the network is another concern.

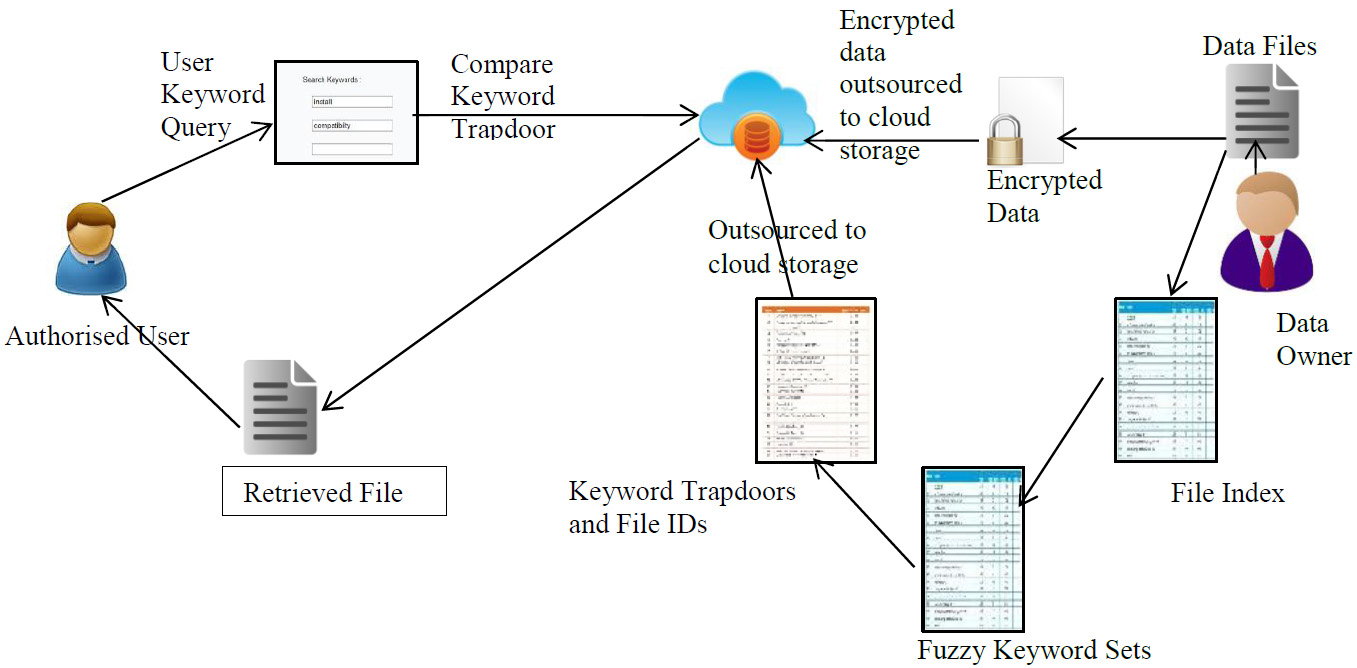
**4. Loss of control**

As the data of client is stored anywhere across the world, control loss over physical, logical of system, and alternative control to client’s assets, mismanagement of assets are some additional concerns.

* 1. **Previous Scenario**

Traditionally the encrypted files along with the key of encryption are stored on same server. However, a system like this means that you need to trust your provider entirely because the provider can access these keys. Even if the people running such a service are entirely trustworthy, the encrypted data can be easily compromised when a private key is accessible by others than yourself alone.

In such a scenario the previously used encryption becomes useless because others now can access all private keys stored on the server to decrypt the data.



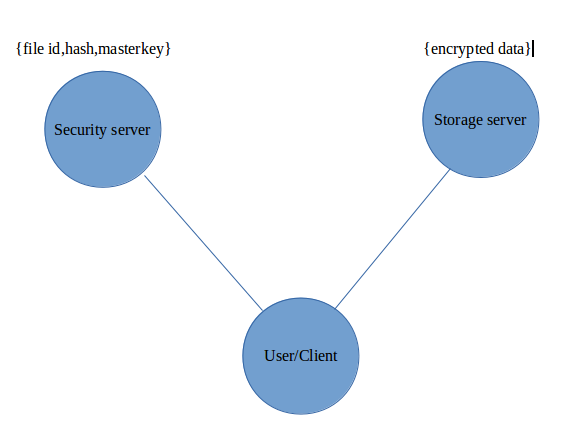
**Fig 2.1 Previous scenario (single server)**

Fig 2.1 shows the use of single server which was the traditional approach not being of great help for secure storage purpose as all the information as well as the key for the data was stored in the single server itself so it was much easier for the hackers to attack this server and exploit as well as retain all the information about the user.

The central server that holds the keys to thousands or even millions of users is of much greater interest to hackers than a personal computer that holds only one or two private keys. Thus, the effort they will put into attacking such a server will also be much higher. Several employees can access these keys. The employees, however, are just people. They can be bribed or blackmailed to turn over private keys, which makes it even easier for hackers or third party agencies to gain access.

* 1. **What we are going to do**

To overcome all above problems we store encrypted data and key on different servers in cloud. As a result even if one server is hacked by the attacker then security wont be affected. Suppose attacker gets access to keys stored on security server he cant do anything with the keys because he don’t have the data. If he gets the encrypted data stored on storage server then also system cant be cracked as hacker don’t have the key to decrypt the data.In any situation security will be achieved.



**Fig 2.2 Current scenario (two servers)**

Fig 2.2 depicts the current scenario that we will be using in our case in order to overcome the problems that occurred in the traditional approach of using single server. So two servers will be used here one is the storage server and the other is security server which makes the work of the hackers difficult as keys and data are stored on different servers and so the hackers cannot exploit or violate the user’s data because the hacker can retain either the key or the data but not both .

**2.5 Cryptography**

Cryptography is a mechanism used for information security and related issues, in particular for encryption and authentication process. During encryption a plain-text is converted into cipher text, while the reverse process termed as decryption converts the cipher text into plain-text. The cipher is in unreadable format[3].

In our approach, we will be using Advanced encryption standard (AES) as an encryption mechanism for encrypting the user’s information before storing it to the cloud, Diffie Helman for key exchange and Secure Hash function (SHA) for hashing purpose.

**2.5.1 Improvements in symmetric encryption**

**DES (Data Encryption Standard)** is a symmetric block encryption standard to be recommended by NIST [3]. The DES algorithm is the most broadly used encryption algorithm in the world. The same algorithm and key are used for encryption and decryption, with minor differences. DES accepts an input of 64-bit long plaintext and 56-bitkey (8 bits of parity) and produce output of 64 bit block.

**AES (Advanced Encryption Standard)** is a symmetric block encryption standard recommended by NIST (National Institute of Standards and Technology) [5] is used for securing information. It uses the same key for both encryption and decryption. . It has variable key length of 128, 192, or 256 bits; default 256 [4]. It encrypts data blocks of 128 bits in 10, 12 and 14 round depending on the key size [7] . Thus AES is a better option as compared to DES as it provides more security and improved performance.

**2.5.2 Improvement in asymmetric encryption**

**RSA (Rivest-Shamir-Adleman)** is broadly used an asymmetric encryption /decryption algorithm which involves a public key and a private key. The public key can be informed to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted using the private key. It secured user data assimilate encryption before to storage, user authentication procedures prior to storage or retrieval, and making secure channels for data transmission [4] . 4096 bit key size is used for execution of RSA algorithm. RSA algorithm involves these steps:

1. Key Generation

2. Encryption

3. Decryption . RSA is very slow for bulk data encryption.

**Diffie-Hellman (DH)** on the other hand, was first revealed by Whitfield Diffie and Martin Hellman in 1976. Diffie–Hellman key exchange is a specific method of exchanging cryptographic keys [6]. It permits two parties that have no prior knowledge of each other to jointly make a shared secret key over an insecure communications channel. This key can then be used to encrypt posterior communications using a symmetric key cipher. Thus, DH is a better choice for asymmetric encryption.

**2.5.3 Improvements in hashing encryption**

**MD5 (Message Digest5)** is a broadly used cryptographic hash function with a 128-bit hash value. It processes a variable-size message into a fixed-length output of 128 bits [3]. The input message is divided into chunks of 512-bit blocks; then the message is padded for making its length divisible by 512[4]. In this sender use the public key of the receiver to encrypt the message and receiver use its private key to decrypt the message.

**SHA (Secure Hashing Algorithm)** is a hashing algorithm. SHA-1 is most extensively used SHA hash function. SHA1 outputs a 160-bit digest of any sized file or input. SHA-256 algorithm produces an almost-unique, fixed size 256-bit (32-byte) hash [6]. This creates it suitable for password validation, challenge hash authentication, anti-tamper, digital signatures. SHA-256 is one of the successor hash functions to SHA-1, and is one of the strongest hash functions available. SHA-256 hash functions computed with 32-bit word.

**2.4 Fuzzy keyword search technique**

A type of search that will find matches even when users misspell words or enter in only partial words for the search.

Techniques used in previous papers are:

1. Wildcard based searching technique[8]

2. N(3) gram search technique[3]

**1. In wildcard based technique:**

, all possible combination of word is produced to compare it with the stored word. For example if the search query is flunk so all possible words like {\*lunk, f\*unk, fl\*nk, flu\*k, flun\*} are produced and matched. Similarly for more number of letters like {\*flunk, f\*lunk,........flun\*k, flunk\*}. So here a lot of words are produced. So the time complexity increases a lot.

**2. N(3) gram based technique:**

fuzzy keyword sets with each set containing words of 3 letters is produced for both the stored word and the search query.

Technique that we are using is N(2) gram based.

The reason for using this technique is :

N(2) gram based technique VS Wildcard technique:

In N(2) gram, sets of words is produced with each set containing words of 2 letter. The advantage of using this technique over Wildcard based is that we do not have to make a lot of comparisons by exchanging a letter from a word. For example for FLUNK the comparison words will be {ALUNK, BLUNK, CLUNK.......FLUNZ). So a lot of words are produced with a lot of comparison which would affect the result time. In case of N(2) gram, only 4 grams are produced{FL,LU,UN,NK}. Using this 4 grams, the approximate result is given as output by comparing the grams of the stored word.

N(2) gram based technique VS N(3) gram based technique:

The advantage of using N(2) over N(3) is that if there is a word with 4 letters or 5 letter like parth, for N(3) the set produced will be {par,art,rth) where as for N(2) the set would be {pa,ar,rt,th}. Now if the search query is typed as paeth, then in case of N(3) the set produced will be {par,aet,eth} and for N(2) it would be {pa,ae,et,th}. So in N(3) no grams are matching whereas in N(2) 2 grams are matching {pa, th}. Therefore in case of N(3) parth wont be displayed as the answer but in N(2) parth will be displayed.

**3. Description**

This section will contain the detail of how exactly the project will work. It consists of what cloud is and its features. It will tell us the scenarios that our project will have like uploading and downloading of data with the features like encryption, decryption and hashing of data. It tells us how the model is secure using the encryption technique AES and lastly how exactly N(2) gram based fuzzy keyword will work.

**3.1 Brief Description**

Generally the cloud services are browser based, therefore any browser enabled device such as for instance laptop, desktop, smart phone, tablets can used to gain access to these services, the services at providers end may be hosted on any platform, from Windows, Linux, etc, which are accessible via internet. As an example consider a regular income and expenditure application which gives different analysis on expenditure by a person, this application could be executing on cloud providers server, whilst the client browser will allow client to feed in the inputs and visualize the analysis prepared for the inputs provided, these analysis computation is completed at server side. Suppose this application can further create documentation on monthly bases which often can be stored in cloud storage once again relieving the client from storing or processing the ﬁle on its side. Because the cloud services are offered via internet, signiﬁcant factors which play an important role in performance are speed of internet, processing power of the individual. While the cloud providers have server banks, to boost the processing power, multiple server are often used internally by the cloud service provides. This pooling is invisible to the client. On another hand if these heavy tasks were to be executed on client side, it would require investment in hardware, time. Due to cloud, it frees the client from buying expensive hardware and investing his/her valuable time, since time is money[5]. Having studied the overview of working cloud, let’s now understand some of the essential characteristics [6].

**3.1.1 On-the-ﬂy service :**

A consumer can require more capabilities at any movement of time, example processing power for huge task, and these requirement must be accomplished without human intervention and be invisible to client [4].

**3.1.2 Wide Accessibility:**

Generally the cloud service are available via standard network protocols, it promotes different types of clients platforms (like, smart phones, laptops etc.) for accessing these services [4].

**3.1.3 Pooling Of Resources:**

The pooling of the resources at cloud providers end is invisible to the end client, and resource assignment is done dynamically depending the need of the client [4].

**3.1.4 Measured service :**

Cloud has enough resources, and amount used by each client is measured by metering capability, and controlled at some level, for optimized resource usage, (like storage).

**3.2 Proposed System**

In the proposed design, a hash service data integrity verification, encryption/decryption service, and provision for defining list of people which can access data securely, is provided by a trusted 3rd party which is separate from the storage cloud provider.

**3.2.1 Business Model with separate encryption/decryption and hashing service:**

The system provides hash, access list, encryption/decryption by a trusted 3rd party over the network in the form of “Software as a Service” (SaaS)[1]. The system has a separate storage service which is also provided as a SaaS. The data storage for each client is done in database in the form of ”BLOB”. The trusted 3rd party which provides these security services does not store any data at its ends, and stores only master key for each client for data encryption and decryption, and hash of the data which is calculated on client side. To enhance the security, the communication between client and security server is secured using Diffie Hellman key, which is used as a input for AES. This division of responsibility has big effect, as no single provider has access to other data and security key, hash at the same time.

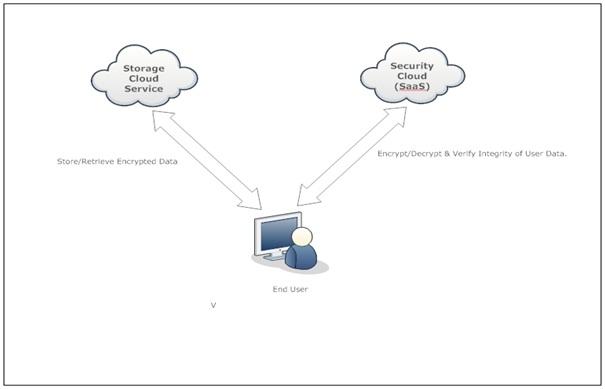
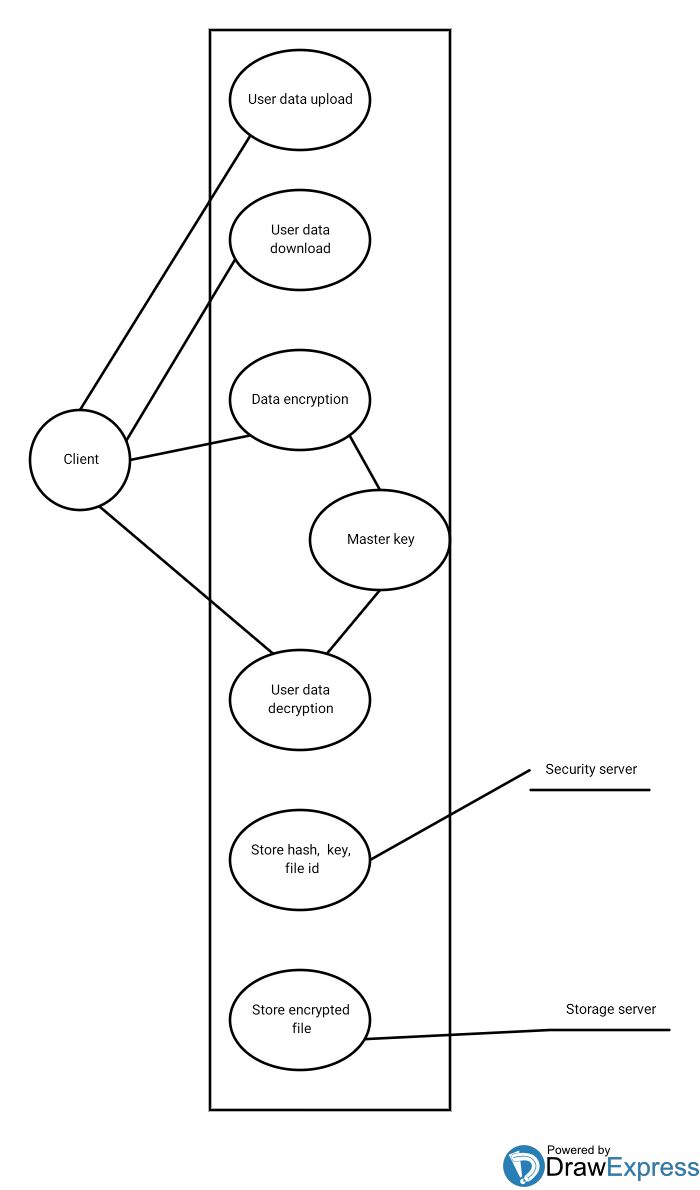


Fig. 3.1: Architecture of data storage and retrieval

Fig. 3.1 is an overview of the architecture where storage and encryption/decryption/hash services (security services) are separated. For example (as described in chapter 1, Motivation) a small or medium scale business who wish to store all its account related data in cloud storage, will first calculate the hash of the data, encrypt the data using encryption service and then store the data in storage provided by separate provider. The system also provides functionality where other users from small scale business Company will be able to access data which is stored in cloud storage. The sessions between client and security server is secured using Diffie Hellman Key and AES as the encryption algorithm. SHA-1 is used for calculating the hash of the data, and AES is used a encryption/decryption algorithm for computing cipher at security server end[7].

**3.3 Methodology and Analysis**



**Fig. 3.2: Use case of data storage and retrieval in cloud**

Fig. 3.1 shows the use case of the data storage and retrieval in cloud. The functions performed are

1. User uploads the data to cloud

2. User downloads data from the cloud

3. Users data is encrypted before uploading

4. A master key is used for encryption of the data

5. Using this master key the data is decrypted when data is downloaded.

6. Security server stores the hash code, master key and file id.

7. Storage server stores the encrypted file.

**3.3.1 User Data Upload Scenario**

1. The end user login to the system with his/her username & password.

2. Once the user is authenticated, the Diffie Hellman key is exchanged for the session.

3. Now a user can select the files which he/she wants to upload it to storage cloud.

4. The hash of the data in file is calculated, using SHA-1 (original hash).

5. The data in file is now encrypted using DH keys.

6. The complete encrypted file and original hash of file data are now transferred to Security Cloud.

7. At Security Cloud, encrypted files is decrypted back using DH key, while the hash is sorted in security cloud database.

8. The decrypted file is now encrypted with Symmetric Algorithm namely AES, using the Master Key generated for each user during user creation.

9. File ID, original hash (file/data hash), master key for each user are stored in Security Cloud database.

10. The Security Cloud now discards any contents of the files from its system, and does not store any file contents in its system.

11. The Encrypted file is sent back to user, to be uploaded to Storage Cloud.

12. The user now can upload the encrypted file to Storage Cloud.

**3.3.2 User Data Download Scenario**

1. The end user login to the system with his/her username & password.

2. Once the user is authenticated, the Diffie Hellman key is exchanged for the session.

3. Now a user can select the files which he/she wants to download it from storage cloud.

4. The encrypted file is now downloaded from storage cloud to user’s match in.

5. The complete encrypted file is now transferred to Security Cloud.

6. The data in file is now encrypted using DH keys.

7. The complete encrypted file and original hash of file data are now transferred to Security Cloud.

8. At Security Cloud, decrypted files with Symmetric Algorithm namely AES using Master Key stored in security cloud database for each user.

9. The decrypted file is now encrypted with DH key.

10. The DH encrypted file and hash of the corresponding file is now passed to the users.

11. At user end, on receiving the encrypted file, it is decrypted with DH keys.

12. The hash of decrypted file is calculated using SHA-1 and original hash are now compared to see if they match, and accordingly appropriate message like, File tampered or File is intact are flashed on user screen. Thus the integrity of the data is verified.

**3.4 Design Approach**

In this section we discuss some of the advantages of algorithms used for encryption (i.e. AES) and used for fuzzy search (N-gram with ranking).

**3.4.1 Selection of Advanced Encryption Standard (AES)**

[9]Broadly speaking the encryption/decryption can be done via symmetric key or asymmetric key. In symmetric algorithms, both parties share the secret key for both encryption/decryption, and from privacy perceptive it is important that this key is not compromised, because cascading data will then be compromised. Symmetric encryption/decryption require less power for computation. On the other hand asymmetric algorithms use pairs of keys, of which one key is used for encryption while other key is used for decryption.

Generally the private key is kept secret and generally held with the owner of data or trusted 3rd party for the data, while the public key can be distributed to others for encryption. The secret key can’t be obtained from the public key. In our case since the encryption/decryption is performed on trusted 3rd party server, symmetric key is used, and it delegates the burden of key management to the trusted 3rd party. If key management where to be done at clients end it would mean,

1. Either they have to remember the big key

2. Store the key in all devices/machine which will be used to access the cloud services, which make user device a bottleneck.

3. Individual owner has to take the responsibility of sharing the key with specific authorized group of user which he/she define.

While on the other hand using symmetric key encryption the master key or private key usage which would be stored in security cloud provider per user gives the client the advantage like,

1. freedom from remembering any key.

2. Client can use any device/machine to access the data stored in cloud.

3. the client need not worry as to how the data will be shared securely, the client just need to define the individual whom he/she wants to share the data with.

**3.4.2 Selection of N-Gram with ranking**

Fuzzy Search: approximate string matching

Ex. Language will be corrected to language

Scenario:

1. User want search keyword language

2. User misspelled it as languaje and clicked on search button

3. Data in the database is in encrypted form.

4. Now we will try to search the encrypted data for inputted keyword languaje. Which will converted to language and display result.

5. This is the technique which will help us to match the keywordlanguaje with encrypted keywords in the database.

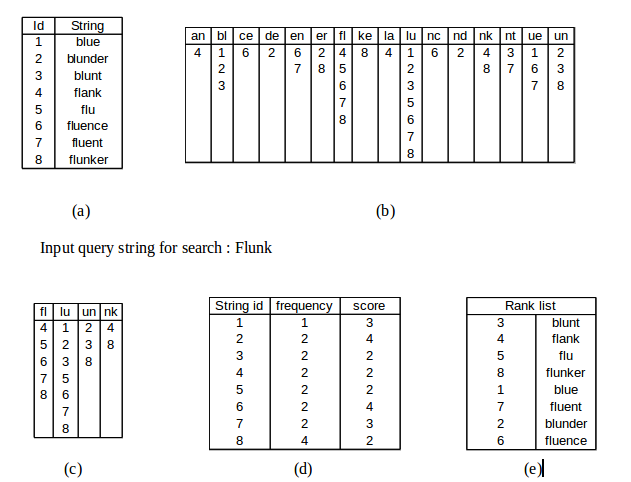


Fig 3.3. (a) List of file name in cloud (b) N(2) gram list of cloud file name

(c) N(2) gram list of searched query (d) frequency and score (e) displayed file

rank-wise

The Fig 3.3 shows how exactly fuzzy keyword search works:

(a) File names like blue, blunder etc. are stored in the cloud.

(b) The N(2) gram list of the files names are formed and each gram forms an index. For example blue is broken as bl,lu,ue and each of it works as an index. Similarly fluent is broken as fl,le,ue,en,nt. With each index all the associated file id are stored. For example for bl index, file id 1 2 and 3 contains bl so we write 1,2 and 3 in the list having bl as index.

Now the search string is given as input.

(c) This contains the N(2) gram list of the search query string with the file ids that match with the grams. For example in flunk, fl is there in 4,5,6,7 and 8.

(d) This consists of the frequency of the grams that match. For example file id 1 matches only 1 time (fl) so the frequency is 1. Similarly file id 4 has frequency 2 (fl and nk). Score is calculated based on the number of grams that are not matched. So in case of file id 1, total grams to be matched was 4 but only 1 matched so the score is 3.

(e) This displays the list of files in rank order. Lower the score of the file, higher is its rank.

**4. Implementation**

This part will talk about the hardware and the software requirements for this project.

**4.1 Software Requirements**

* Java - to code fuzzy and encryption algorithms
* Web Service - needed to implement web service
* SOAP API - to be able to call web service at client side we need to use SOAP API or even XML version 3.2.
* OpenStack - cloud deployment service

**4.2 Hardware Requirements**

* OS - Windows or Linux
* RAM - 2.0 GB or greater

**5. Further Work**

In semester 8, we will implement the proposed system by creating two servers (storage and security server). Storage server will store all the data and security server will store the master key for encrypting the user data. We will also implement fuzzy keyword search technique to improve the performance of the system.

**6. References**

[1] L. Liu, C. Zhang, S. Yao, S. Wang, and W. Zhou, "Fuzzy Keyword search with safe index over Encrypted cloud computing," TELKOMNIKA Indonesian Journal of Electrical Engineering, vol. 11, no. 10, Oct. 2013.

[2] N. Smetanin and V. my complete profile, "Fuzzy string search," 2011. [Online]. Available: http://ntz-develop.blogspot.in/2011/03/fuzzy-string-search.html. Accessed: Nov. 15, 2016.

[3] A. Meharwade and G. A. Patil, "Efficient Keyword search over Encrypted cloud data," Procedia Computer Science, vol. 78, pp. 139–145, 2016.

[4] H. Bast and M. Celikik, "Efficient fuzzy search in large text collections," ACM Transactions on Information Systems, vol. 31, no. 2, pp. 1–59, May 2013.

[5] J. Wang et al., "Efficient verifiable fuzzy keyword search over encrypted data in cloud computing," Computer Science and Information Systems, vol. 10, no. 2, pp. 667–684, 2013.

[6] Z. Fu, X. Sun, Z. Xia, L. Zhou and J. Shu, "Multi-keyword ranked search over Encrypted cloud data supporting synonym query," International Journal of Science and Research (IJSR), vol. 5, no. 6, pp. 2044–2048, Jun. 2016.

[7] W. Zhou, L. Liu, H. Jing, C. Zhang, S. Yao, and S. Wang, "K-gram based fuzzy Keyword search over Encrypted cloud computing," Journal of Software Engineering and Applications, vol. 06, no. 01, pp. 29–32, 2013.

[8] N. Shekokar, K. Sampat, C. Chandawalla, and J. Shah, "Implementation of fuzzy Keyword search over Encrypted data in cloud computing," Procedia Computer Science, vol. 45, pp. 499–505, 2015.

[9] P. null, "Enhanced privacy-preserving Multi-Keyword ranked search over Encrypted cloud data," American Journal of Networks and Communications, vol. 4, no. 3, p. 25, 2015.

**7. Acknowledgement**

Any accomplishment requires the efforts of many people and our project is no different.

We would like to thank our internal guide **Prof. Shachi Natu** for her overwhelming support during the entire education of our project and for being the task-master and the mentor. We would like to express our gratitude towards him for her constant encouragement, support and guidance throughout the project.

It is only right to express our sincere gratitude to our Professors, BE IT Batch of 2017, our classmates whose help and support were instrumental and essential in the development and successful completion of this phase of our project. We also thank the teaching staff and non-teaching staff of the IT Department who have helped us from time to time with necessary resources and otherwise deserve a special mention.

Ultimately, we would like to thank the open-source community, world over, who have taken painstaking efforts and provided us with the useful information available on the Internet.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Saumya Sharma (136) Parth Agarwal (137)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Amrita Bhagtani (138) Ankit Mohite (139)