# Implementation of Fuzzy Keyword Search Using Encryption & Decryption over Cloud Computing

Prof. S.S. Vanjire,
Professor at Dept. of CE, Sinhgad Academy of Engineering, Kondhwa, Pune-48
Swapnil Bhapkar[1],
Amardeep Bikkad[2],
Gaurav Bobade[3,] Dhiraj Bandle[4]
1,2,3,4,- Students, Department of CE, Sinhgad Academy of Engineering, Kondhwa, Pune,
Savitribai Phule Pune University, Pune, India

**ABSTRACT**: With the increased rate of growth and adaption of cloud computing, daily, more and more sensitive information is being centralized onto the cloud. For the protection of valuable information, the data must be encrypted before externalization. The existing search techniques allow the user to search over encrypted data using keywords but these techniques account for only exact search. There is no tolerance for typos and format inconsistencies which are normal user behaviour. This makes effective data storage and utilization a very challenging task, supplying user searching very frustrating and inefficient. In this paper, we focus on secure storage using Advanced Encryption Standard (AES) and information retrieval by performing fuzzy keyword search on data which is encrypted while uploading on the cloud. We are proposing the implementation of an advanced fuzzy keyword search mechanism called the Wildcard based techniques which returns the matching files when users searching inputs precisely match the predefined keywords or the closest possible matching files based on similarity keyword semantics when exact match fails. In the proposed solution, we utilize edit distance to quantify keywords similarity and develop an efficient techniques for constructing fuzzy keyword sets, which focus on reducing the storage and representation aloft.

***KEYWORDS:*** Fuzzy search, encryption on cloud, cloud computing, AES, wildcard.

## I. INTRODUCTION

Cloud computing is an essential platform for sharing resources. This sharing of resources is based on three models: Infrastructure-as-a-service (IaaS), Platform-as-a-service (PaaS), Software-as-a-service (SaaS).These services are customized as per user demand. Cloud computing more preferably referred as "the cloud", also focuses on maximizing the effectiveness of shared resources. Usually cloud resources are not only shared by multiple users but are also reallocated as per the user demand. The storage of data on cloud reduces the burden of storage and maintenance of data on the user. Taking into account this tremendous growth of sensitive information on cloud, cloud security is of vital importance for enterprises. The fact that the information owners and vendor provisioned cloud servers are not a part of the same trusted domain may put the outsourced data at risk. This growth of cloud servers are not a part of the same trusted domain may put the outsourced data at risk. This growth of cloud service users has unfortunately been accompanied with a growth in malicious activity in cloud. More and more vulnerabilities are being investigated nearly every day. Millions of users are subscribing to the cloud based services, therefore safety and security of these services are of utmost importance. The future of cloud, even more in expanding range of applications, involves a much deeper degree authentication and privacy. Proposed here is a simple data protection model where data is encrypted using Advanced Encryption Standard (AES) before it is launched into the cloud, thus ensuring data confidentiality and security.

To ensure security during information retrieval, we are employing a searchable encryption mechanism.in a standard searchable encryption scheme, an index is created for every keyword of interest and it is associated with the files that contain the keyword. The trapdoors of the keyword are integrated with the index information, thus effective keyword search is realized without compromising file content. In an age of intelligent search systems, the standard searchable encryption scheme supporting an exact keyword match is inconsistent with casual user search behaviours. Normal user search queries will have typos and representation irregularities which may not match the pre-set keyword strings. A user searching for "APPLE" can accidentally type "APLE" and another person may query for "P.O.BOX" because he is ignorant about the stored keyword.

Thus, we shift our focus on enabling effective privacy-preserving fuzzy keyword search for information stored in cloud environments. Fuzzy keywords search augments system usability by returning the matching files when users searching inputs exactly match the pre-defined keywords or the closest possible matching files based on keyword similarity semantics, when exact match fails. Edit distance is used to quantify keywords similarity and for the development of the novel technique, i.e., a wild-card based technique, for constructing fuzzy keyword sets. This technique eliminates the need for counting all the fuzzy keywords and the total size of the fuzzy keywords set is significantly decreases.

## II. RELATED WORK

AES is a block cipher that has block length of 128 bits. Three different key lengths are allowed in it i.e. 129, 192, or 256 bits. We propose AES with 128 bit key length. The encryption process is of 10 rounds of processing for 128-bit key. Except of the last round in each of the case, all other rounds are identical.16 byte encryption key, in the form of 4-byte words is expanded into a key schedule consisting 0f 44 4-bytes words. The 4X4 matrix of bytes made from 128-bit input block is referred to as the state array. Before any round-based processing for encryption can begin, input state is XORed with the first four words of schedule. The importance of fuzzy search has received attention in the realization of plaintext searching for information retrieval. This problem was addressed by allowing user to search relevant information based on approximate string matching. It seems possible for one to directly apply these string matching algorithms to the context of searchable encryption by computing the trapdoors on a character base within an alphabet. However, this simple construction suffers from dictionary and statistical attacks due to lack of privacy-preserving encryption methods. Among the searchable encryption techniques, most of those works are focused on efficiency improvements and formalization of security definitions. Searchable encryptions first construction was proposed by song at al, in which each word in the document is encrypted independently under a special two layered encryption construction. Goh proposed to use bloom filters for constructing the indexes for data fiels.to achieve more efficient search, Chang et al And Curtmole et al both proposed similar approaches in which a single encrypted hash table index is built for the whole of file collection. In this each entry consists of the trap-door of a keyword and an encrypted set of file identifiers whose corresponding data files contain the keyword. A complementary approach was presented by Boneh el al as a public key based searchable encryption scheme.

## III. PROPOSED WORK

Cloud Computing is a construct that allows you to access applications that actually reside a remote location. Cloud computing uses the internet and central remote servers to maintain data and applications; the data is stored in off-premises and accessing this data through keyword search. Encryption on data in cloud is done using the Advanced Encryption Standard (AES) algorithm. The user decides to cloud services and outsource his data on the cloud. User submits his service requirements with Could Service Providers (CSPs) and chooses the provider offering best specified services. To fully exploit potential of cloud computing there should be limited restrictions on processing and computation. This possible when we enable encrypted data search. For this, there exists a model where CSP's can partially access the data without having to undo the encryption. Updating, querying or sharing a dataset without leaking any information to the cloud provider is possible.

### *Implementation of fuzzy keyword search*

We propose a scenario where a private enterprise would like to centralize its data storage to 5s5cloud.The enterprise data files are encrypted using AES and outsourced to cloud storage. At the same time the following information is stored in a FILE INDEX: a) File ID. b) File name. c) Keywords. We derive out fuzzy keywords sets from this FILE INDEX using Edit Distances and Wild-Cards section. These fuzzy keyword sets associated with their respective file identifications. On the Fuzzy Keyword Sets generated, the trap-door function is applied. The keyword trapdoors and File IDs are now outsourced to cloud storage. Now, the enterprise adds users who are authorized to access their data. The user enters a search query for file retrieval. The trapdoor function is applied on the search words and compared with the existing keyword trapdoors. The relevant matching files are received and user selects his/her file of interest. The selected file is then decrypted and made available for the user.

### METHODOLOGY

### 1. Wildcard Based Fuzzy Set Construction

In a straightforward approach, all the variants of the keywords have to be listed even if an operation is performed with multiple instances on one position. Based on this, we proposed to use a wildcard to denote edit operations at the same position. The fuzzy set (based on the wildcards) of wi with edit distance d is denoted as S wi,d={S 'wi, S' wi,1, …, S' wi,d}, where S' wi,t denotes the set of words W' I with t wildcards. Here, every wildcard represents an edit operation on wi. For example, for the keyword APPLE with the pre-set edit distance of 1, its fuzzy keyword set based on wildcards can be constructed as Sapple,1 = {APPLE, *APPLE, *PPLE, A*PPLE, A*PLE, APPL*E, APPL*, APPLE*}. The total no of variants on APPLE constructed on this way is only 13 + 1, instead of 13 x 26 + 1 as in the above exhaustive enumeration approach when the edit distance set as 1. For a given keyword wi, generally, with length l, the size of Swi,1 will be only 2l +1+1, as compared to (2l + 1) x 26 + 1 obtained by using the straightforward approach. Larger the pre-set edit distance, more the storage overhead which can be reduced: with the same setting of the example as demonstrated in the straightforward approach, the proposed methodology can help reduce the storage of the index from 30GB to down to near 40GB. In case of the edit distance being set to 2 and 3, the size of Swi,2 and Swi,3 will be Csqr1l + 1 + Csqr1l * 2Csqr2l + 2 and Csqr1l + 1 + Csqr3l + 2Csqr2l + 2Csqr2l * Csqr1l. In other words, the number is only O(lsqrd) for the keyword with length l and edit distance d.

**2. The Efficient Fuzzy Keyword Set Construction Scheme**

Based on the storage-efficient fuzzy keyword sets, we show the construction of an efficient as well as effective fuzzy keyword search scheme. The fuzzy keyword search scheme goes like:

1) To build an index for wi, having an edit distance d, the data owner must first construct a fuzzy keyword set Swi,d using the wildcard based technique. Then he computes trapdoor set {Tw'i} for each w'i E Swi,d with a secret key sk shared between data owner and authorized users. The data owner encrypts FIDwi as Enc( sk,FIDwi || wi). The index table {({Tw'i}w'i E Swi,d, Enc(sk, FIDwi || wi ))}wi E w and encrypted data files are outsourced to the cloud server for storage.

2) To search with (w, k), the authorized user computes the trapdoor set {Tw'}w'ESw,k, where Sw,k is also derived from the wildcard-based fuzzy set construction. He then sends {Tw'}w'ESw,k to the server.

3) Upon the receiving the search request {Tw'}w'ESw,k , the server compares them with the index table and returns all the possible encrypted file identifiers {Enc(sk,FIDwi || wi)} based on the definitions of the fuzzy keyword. The user can then decrypt the returned results and retrieve relevant files. The technique of constructing search request for w is the same as the construction of a keyword index. Thus, the search request is a trapdoor set on the basis of Sw,k, instead of a single trapdoor of the straightforward method. Going in this manner, the searching result correctness can be ensured.

## IV. SIMULATION AND RESULTS

Test Case Scenario 1: Registration

| Test Case ID | Test Scenario | Test Steps | Test Data | Expected Result | Actual Result | Pass/Fail |
|---|---|---|---|---|---|---|
| TC1.1 | Valid Data | 1.Go to Webapp<br>2. Enter user details<br>3.Click Submit | register | User/ vendor should be registered. | As expected | Pass |
| TC1.2 | Invalid Data | 1.Go to Webapp<br>2. Enter user details<br>3.Click Submi | register | User should not register and error message should popup. | As expected | Pass |
| TC1.3 | Empty fields | 1.Go to webapp<br>2. Click Submit | register | Error message should pop up | As expected | Pass |

Test Case Scenario 2: LOGIN

| Test Case ID | Test Scenario | Test Steps | Test Data | Expected Result | Actual Result | Pass/Fail |
|---|---|---|---|---|---|---|
| TC2.1 | Valid Data | 1.Go to webapp<br>2. Enter user id<br>3. Enter password<br>4.Click Submit | Login:<br>User Id<br>Password | User should login in. | As expected | Pass |
| TC2.2 | Invalid Data | 1. Go to webapp<br>2. Enter invalid User id<br>3. Enter invalid password<br>4. Click Submit | Login:<br>User Id<br>Password | User should not login and login page should appear again with empty fields for login. | As expected | Pass |
| TC2.3 | Empty fields | 1.Go to webapp<br>2. Click Submit | Login:<br>User Id<br>Password | User should not login and login page should appear again with empty fields for login. | As expected | Pass |

Test Case Scenario 3: User activity

| Test Case Id | Test Scenarios | Test Steps | Test Data | Expected Result | Actual Result | Pass/Fail |
|---|---|---|---|---|---|---|
| TC3.1 | Select file | Browse And select | To select a file | Vendor details stored successfully | As expected | Pass |

| | | | | | | |
|---|---|---|---|---|---|---|
| TC3.2 | Fill keyword | Enter keywords | To input keywords for fuzzy search | Keywords stored | As expected | Pass |
| TC3.3 | encrypt | Click on encrypt | To encrypt selected file | Encryption successful | As espected | Pass |
| TC 3.4 | Upload file | Upload on cloud | To upload encrypted file on cloud | Upload successful | As expected | pass |
| TC 3.5 | Search using keyword | Enter keyword | To search For a file using a keyword | File displayed | As expected | pass |
| TC 3.6 | Decrypt and download | Click on decrypt | To decrypt and download searchd file | Decryption and download successful | As expected | pass |

## V. CONCLUSION AND FUTURE SCOPE

In the future scope of this system we are willing to do the indexing of the mapped words and fuzzy sets so as to increase the functionality of the search procedure. Encryption of more file formats can be done. Also decryption of image and media files can be done. In this paper we aim to make a privacy-preserving fuzzy search for achieving effective usage of remotely stored encrypted data in clouds computing. We are designing an advanced search algorithm mechanism (i.e. Wildcard Based Technique) for constructing storage efficient fuzzy-keyword sets based on the similarity metric edit distance. Based on the fuzzy-keyword sets, we propose a fuzzy keyword search technique.

## VI. REFERENCES

[1] Google, “Britney spears spelling correction,” Referenced online at http: //www.google.com/jobs/britney.html, June 2009.

[2] M. Bellare, A. Boldyreva, and A. O’Neill, “Deterministic and efficiently searchable encryption,” in Proceedings of Crypto 2007, volume 4622 of LNCS. Springer-Verlag, 2007.

[3] D. Song, D. Wagner, and A. Perrig, “Practical techniques for searches on encrypted data,” in Proc. of IEEE Symposium on Security and Privacy’00, 2000.

[4] E.-J. Goh, “Secure indexes,” Cryptology ePrint Archive, Report 2003/216, 2003, http://eprint.iacr.org/.

[5] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, “Public key encryption with keyword search,” in Proc. of EUROCRYP’04, 2004.

[6] B. Waters, D. Balfanz, G. Durfee, and D. Smetters, “Building an encrypted and searchable audit log,” in Proc. of 11th Annual Network and Distributed System, 2004.

[7] Y.-C. Chang and M. Mitzenmacher, “Privacy preserving keyword searches on remote encrypted data,” in Proc. of ACNS’05, 2005.

[8] R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, “Searchable symmetric encryption: improved definitions and efficient constructions,” in Proc. of ACM CCS’06, 2006.

[9] D. Boneh and B. Waters, “Conjunctive, subset, and range queries on encrypted data,” in Proc. of TCC’07, 2007, pp. 535–554.

[10] F. Bao, R. Deng, X. Ding, and Y. Yang, “Private query on encrypted data in multi-user settings,” in Proc. of ISPEC’08, 2008.

[11] C. Li, J. Lu, and Y. Lu, “Efficient merging and filtering algorithms for approximate string searches,” in Proc. of ICDE’08, 2008.
[12] A. Behm, S. Ji, C. Li, , and J. Lu, “Space-constrained gram-based indexing for efficient approximate string search,” in Proc. of ICDE’09.

[13] S. Ji, G. Li, C. Li, and J. Feng, “Efficient interactive fuzzy keyword search,” in Proc. of WWW’09, 2009.

[14] J. Feigenbaum, Y. Ishai, T. Malkin, K. Nissim, M. Strauss, and R. N. Wright, “Secure multiparty computation of approximations,” in Proc. of ICALP’01.

[15] R. Ostrovsky, “Software protection and simulations on oblivious rams,” Ph.D dissertation, Massachusetts Institute of Technology, 1992.

[16] V. Levenshtein, “Binary codes capable of correcting spurious insertions and deletions of ones,” Problems of Information Transmission, vol. 1, no. 1, pp. 8–17, 1965.