DOI: http://dx.doi.org/10.26483/ijarcs.v9i2.5912

ISSN No. 0976-5697

Volume 9, No. 2, March-April 2018

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

SURVEY PAPER ON FILE SEEKING OVER ENCRYPTED OUTSOURCED DATA

Brahm Datt
Department of Computer Science and Engineering
Lovely Professional University
Phagwara, Punjab, India

MrinaliniRana
Department of Computer Science and Engineering
Lovely Professional University
Phagwara, Punjab, India

Abstract: Over the year People have become more privacy conscious. Security require constant effort, one simply can't rely on just some top notch antivirus or firewall software's. So, best solution is to encrypt the data before sending it to any of the third party storage. Now, in this scenario where user choose to encrypt all the data before outsourcing; what happen is performing regular operations like search becomes a very hectic task; as one has to decrypt all data before actually starting the searching process. In this paper, we are exploring a technique that will help user perform search over encrypted data. Also the main attraction of this technique is even though the user can't remember the exact keyword to look for the particular encrypted file;still the user will be able to search with partially correct keywords. We will also discuss some string matching algorithms in this paper.

Keywords: cloud computing, catchphrase, keyword search, wild card based search, q-gram, n-gram, hashing, encryption search.

1. INTRODUCTION

In this epoch age of digitization where everything and everyone is using technology to store and process all kinds of data, whether for personal use or to satisfy some professional curiosity or some other needs. Almost everyone is creating or updating digital data or at least have some digital presence in form of data, multimedia or information files. And this data growth is, as rapid as twice per year. There is human generated data as well as machine generated data which is encountering a general ten times more data spreading rate than conventional business information, also machine information is expanding significantly more quickly at 50x the development rate. Now in this area of vast data, cloud computing is a crucial feature.

A. Cloud Computing

Cloud computing in twenty first century has been providing remarkably excellent services for all kinds of business, professional, and personal use. Five of the leading upsides of using cloud computing are:

- Self-benefit Catering: End consumers can turn up figure assets for as per scope of work.
- Adaptability: Companies can scale up as, needs increment.
- Pay per utilization.
- Workload strength.
- Migration adaptability.

Now a cloud service provider will most likely be a third party. So, a user can encrypt and save the data on cloud with an ease that third party cannot know about the user data. Now to perform easy plain text like operation on encrypted data Goh [1] represented a secure index technique which was pretty good. Also Y. Li in[2]showcase another scheme in which, a technique called fuzzy keyword search over cloud data or outsourced data was used. By utilizing fuzzy multi keyword search the ease of use for searching over encrypted data is upgraded. Clients can seek their content with conceivable values and get the desired outcome but in situations where user is unable to remember the correct catchphrase for finding the files. This is the point where fuzzy multi keyword search shows its beauty and give wanted outcomes to the client. Even in the case where user did few spelling or typos mistakes. Fuzzy logic provide great advantage in this type of problem as it reduce variable values to lie in between 0 and 1.

B. Fuzzy Logic

Fuzzy logic is a way to cope up with processing of unquantifiable data like one would say "best food"; now what does best food quantify as? It's nothing in terms of computer data processing one cannot use this. So, for this type of data where one can't simply say yes (1) or no (0). Fuzzy logic paves the way and provide a mathematical way to quantify this data into 1's and 0's.Boolean rationale on which the computer works. Fuzzy logic narrows down the computation in way as close to human thinking as possible.

C. Fuzzy Keyword Search on Encrypted Data

In simple words in to search over encrypted data an index is created of every file that is being stored on the cloud and after that the file is encrypted and stored in the cloud. Also file information which used to search particular file or document are stored in the form of n-grams or wild card (for eliminating typos and spelling mistakes). Now whenever a user want to search something like file named "panda"now this keyword is compared with the stored wild card index values of various encrypted files. And whenever a match found corresponding file is decrypted and sent to the user. So, this technique was improved for the misspelled keyword search and ranked keyword search and various other schemes were proposed to accomplish user friendly and efficient searches on the encrypted data.

Now in past years several schemes have been proposed for searching which we discuss in II section and III section contains comparative analysis and result graphs and IV section concludes this paper.In this paper we have done

some past research Study which is written in below section II.

2. LITERATURE SURVEY

M.Sharma[3]in his paper examine different search techniques as: "Index searching using hashes of encrypted data". Workload is mainly done by the server so it give fast result. Also have a security flaw that is Server can get an idea of which files have which number of data as the plain file as well as encrypted file are of same length.

Trapdoor technique is another examined technique in M. Sharma's paper; it is unique in a way that it encrypts data with a keyword and key attached. The attached key acts as a key to perform look-up operation. Memory space are not needed as much as hash indexed of same file size. Except for the keyword encryption and one hash operation rest work is done by server. Except for the data owner no one can know the data element.

Secret sharing technique using multi party communication uses two polynomial tree of XML data to share data securely. Storage is used similar to the plain text in the form of a tree. The workload is identical to that of the server on the client side. Search is supported by joined data of server and client polynomial tree of actual data. Highly secured as the server store all scrabbled data and without client sub-tree it's all useless.

Z. Fu's [4] paper proposed a color property based multi look over encoded cloud OOXML information. Utilize two vectors for the benefit of the keyword and the relating coding of color. Memory require to store two vector data related to keyword as well as the color of keyword. Also results user query with group of fine result files. Search type uses not only keyword but also matches the color Apart from the data the index tree itself is being used in encrypted form.

2016 published this paper [5]consider and take care of the issue of customized multi-keyword graded look over encoded information while protecting security in the outsourced data computing. Uses wordnet for word ranking so additional storage required. Apart from user search history and recent data access. Output result are based on the user search keyword priority as well as the related keyword with help of wordnet. Search is based not only on extracted keyword but on the access frequency given to each keyword. Moreover this paper Examine and prohibit server from realizing any data by known encrypted file and portions of plain files.

In another research a new [6]scheme is devised a semantic search technique utilizing conceptual graphs. Uses 3 vector for storing each file's conceptual graphs. Memory requirement include conceptual graph (CG) generation same as of file size as well 3 vector index for each CG. Uses low communication and computation overhead as most work is done on the server itself except for trapdoor generation and cyphering. Uses semantic Search using conceptual graph while also maintaining fine ranked result. Uses a trapdoor to ensure files as well as index security.

[7]Devised a look-up technique utilizing concept hierarchy along with semantic assist. That is while user query using trapdoor two vector comparison are done one for matching file's index stored concept with user keyword and second whether the concept satisfy query. Search result are more satisfactory to the query matching document concept with semantic support.

C. Hu in [8] his research paper proposed another special case for performing look up on encrypted data. In their technique, they construct one set for each extracted word stored in index, where in the set they recorded each character of this index stored word but without wild card. Also for this technique they use the normal character order in the keyword, reverse order and the alphabet available.This strategy is extremely effective yet has some little mistake probability. The research paper further explored the construction of Bloom filter channel for every stored index word using wild card. Also, it can be used to a dynamic structure to accomplish security against versatile picked catchphrases security breaches (CKA-2 security).

T. Suga in his [9] work, utilized Pseudo Random Function (PRF) for security change. The PRF can hide the data on the grounds that no proficient calculation method can recognize a solution of PRF from a solution of random functions. Moreover it utilize the Bloom filter to diminish the information size. Bloom filter is used to store all characters of one catchphrase. Furthermore utilize PRF while adding a character to Bloom filter and symmetric key encryption method to encrypt information for the search items.

X. Zhu in [10]his paper used an uc (universal composability) technique by kurosawa et al. called sse scheme to detect the deceptive behavior of the server. To further explain the concept of search verification that whether it's coming from trusted server or not. They proposed their search technique to be verified byuc-secure. The UC-Secure is the security of a protocol proven in a standalone setting is preserved under composition if it is secure in the universal composable framework.

S.A. Mittal [11]explores multi-keyword and ranked search. Thus it makes the encrypted data search closer to the normal text like search.Additionally it introduces the index generation for ranked search. In contrast it explores synonym based search plus ranking them for faster access.

Z. Fu [12]explains and overcome some previous technique used for search over encrypted data. Like use of bi-gram.As it increase very high inconsistency for Euclidean distance calculation for matching misspelled word.Uni-gram based technique is used to reduce distance for Euclidean calculation. For one letter misspelled as well as composite letter errors.Additionally use of stemming algorithm for search keyword with same root.Constructed a keyword file based on the weight of the keyword. That is the files relevant to keyword has greater chance to appear.

Zhao in[13] paper explores the possibility of the fuzzy keyword search over the probabilistic xml data. The probabilistic xml data consists of the label tree with ordinary and distributed nodes. In simple terms the data can be said pretty scattered that is not related or from different sources, which makes it hard to process.What was interesting in this paper was a pruning technique (defined outset value for word probability in index)is used for result refinement using Fuzzy keyword search on related inverted index.

W. Ding[14] using his Chinese keyword secure search explores the fuzzy keyword search over encrypted data for searching the keyword in Chinese words.But itdo shine some light on some interesting techniques Keac (automatic keyword extraction) andPat –tree.It also tells that similar keyword search is more efficient if normal matching fails.

But for that to work we need to generate the similar keyword. But it focuses on the improvement for Chinese auto keyword generation using pin-yin technique. But it certainly gave the new or widened the perspective for searching using similar keyword.

W. Jie[15] aimed to maintain the accuracy of the result in regard of the keyword inputted.It construct a pointer vector for ranked keyword search by using the user feedback.Also they show that their fuzzy set construction is efficient using mathematical model.Not just the one input keyword is compared for the result but also the fuzzy set create a keyword set to be matched for the result.

H. Tuo[16]Well this paper uses the bloom filter for reducing the cost search over encrypted data.Bloom filter uses individual hash functions to map every element to a random no uniform over the range of data.So it reduces the number of compares to be done and speed up the search process.

Q. Xu[17]proposes technique comprised by two technique Fuzzy keyword search andRanked keyword search over encrypted data. Which they show as using more efficient utilizing ranked keyword for gives more accurate and close to the user query outturn.

L. Xue[18] shows that k-gram technique and its shortcoming were identified and omitted.As in k-gram technique not setting k value to right amount leads to drastically poor result. So, a rough fuzzy set is proposed to cover the keyword correctness.

What they actually done was that proposed to check the inputted keyword correctness for search operation should further continue or not.

Bing Wang [19] sets very direct rules of privacy maintenance for retrieving the data from the cloud. It creates the index on an algorithm called bloom filter which helps in search. This paper proposed a technique which calculates the similarity in user keyword and encrypted document by measuring the uncertainty between query and index and result selective matching index only.

3. PRE-RESEARCH WORK

While doing the research study we also devise a comparison survey of several string matching algorithms which can be embed to enhance the performance of search query while performing search on the indexed data. There are various string matching algorithms available many of them are language specific too; But for this comparison survey we devise a list of following algorithms which are not limited to same string length string matching rather can perform various operations and calculations to match two strings for similarity.

In thefollowing Table I,we illustrate the various insertion, substitution, deletion and transposition they use. The string matching algorithms which uses these operation or combination of operation are shown with their respective operations in the below TableI

Table I LCS = Longest Common Substring distance, OSA = Optimal String Alignment a.k.a Restricted Damerau-Levenshtein distance, LV = Levenshtein, DL = Damerau-Leveshtein

String Matching	Operations			
Algorithm	Substitution	Deletion	Insertion	Transposition
LCS	No	Yes	Yes	No
LV	Yes	Yes	Yes	No
OSA	Yes	Yes	Yes	Yes
DL	Yes	Yes	Yes	Yes

1. **Levenshtein Distance (LV):** This string matching algorithm tells if two strings are matched based on Minimal number of additions, erasures and swaps required for changing string X into string Y.
2. **Full Damerau-Levenshtein distance (DL):**This algorithm apart from the insertion deletion and substitution of single character also allows the transposition of contiguous characters are permitted.
3. **Restricted Damerau-Levenshtein distance (OSA):**This algorithm is like Full Damerau-Levenshtein yet every substring are allowed to be altered once only.
4. **Longest Common Substring Algorithm (LCS):**This string matching algorithm utilize the method that least number of alphabets that must be expelled in the two strings until the point that subsequent substrings are indistinguishable. Say for example LCS Distance("Sound", "Hound") = 2i.e removing S and H will leave "ound", "ound" which both are indistinguishable.
5. **Q-gram distance:** Sum of total contrasts between N-gram vectors of the two strings. While applying q-gram the measure of the N-gram; must be non-negative. Like for example-
   Q-gram Distance for ("Sound" and "Hound") will give result 2 as:

Table II Q-gram example

$gram$	So	ou	un	nd	Ho
V1	1	1	1	1	0
V2	0	1	1	1	1

We take q-gram size = 1,2, 3 for this research study purpose. Also q gram for use in other algorithm is also taken as q = 1, 2, 3.

6. **Cosine Distance for Q-Gram Count Vectors:**This algorithm provides the string matching result by comparing the both N-gram vectors for given strings.It calculates the similarity as:
   $$1 - |X \cap Y| / |X \cup Y|$$
   Where X will contain the distinguish q-grams of the string one and Y will contain the string two's q-gram.
7. **Jaccard Distance for Q-Gram Count Vectors:**This algorithm provide similarity by
   $$Jaccard = 1 - x/y$$
   Where x = shared q-grams in strings and y = q-grams in either strings.
8. **Jaro Distance and Jaro-Winkler Distance (JW):**This algorithm is defined as a similarity between two strings by resulting a value range from 0 -1. Where 0 being totally matched case and 1 being the totally different string.It is calculatedas:

$$(1-\left(\frac{1}{3}\right)\left(\frac{w1m}{|s1|}+\frac{w2m}{|s2|}+\frac{w3(m-t)}{m}\right))$$

Where wi = weight assigned to letters of s1, |s1| = number of letters in s1, |s2| = number of letters in s2, m = number of letters matched, t = number of transposition of matching letters.

The jarowinkler distance[20] append a valid range for p is$0.25 \geq p \geq 0$. It's a rectification term to the Jaro-Distance. It is characterized as d-l*p*d, where d is the Jaro-distance. Here, l is gotten by tallying, from the beginning of the information strings, after what number of characters the primary character vary between the two strings appearance, with a greatest of four. The factor p is a punishment factor, which in Winkler is regularly picked 0.1.

For comparison of these algorithms we prepared a list of best average and worst cases:

Table III String table used for comparative analysis; black color represent original string, red = worst; green=best; yellow=average

"Hound Hunting"	
'Hound Hunting'	'Huonthe'
'Bound Hunting'	'Houdhuning'
'Huond Hunting'	'Running Hares'
'Hund Hunting'	' Spectacle'
'Hound O. Hunting'	'Pivotal Moment'
'Hunting, Hound'	'Beyond Relentless'
'HouningHuntd'	'Spectacle Sight'
'BounddHnutin'	'peaceful balnce'
'Detective Hound Hunting'	'News Hound'
'H.o.u.n.d. .H.u.n.t.i.n.g.'	

Best Case: When comparing string is exactly same as the given string or only have 1 character alteration. We take this as the best case scenario. Represented by green color.

Average Case: Where the string is permuted either within in between character or the first and last character. Average case also include the typos character which may or may not happen due to user error. Represented by light yellow color.

Worst Case: The strings that are totally different are considered as the worst case. Represented by red color.

Comparison is performed by comparing all case output for each algorithm with varying variable in case of jaccrad, cosine, q-gram and jw.

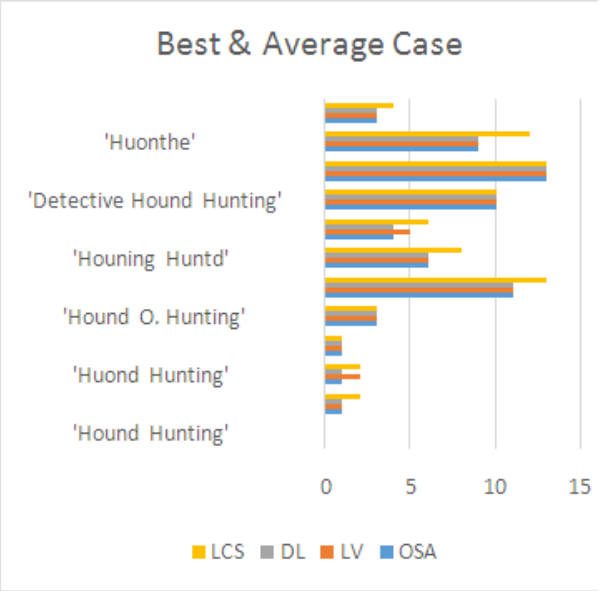

Figure I Difference chart of OSA, LV, DL, and LCS with close to 0 means match

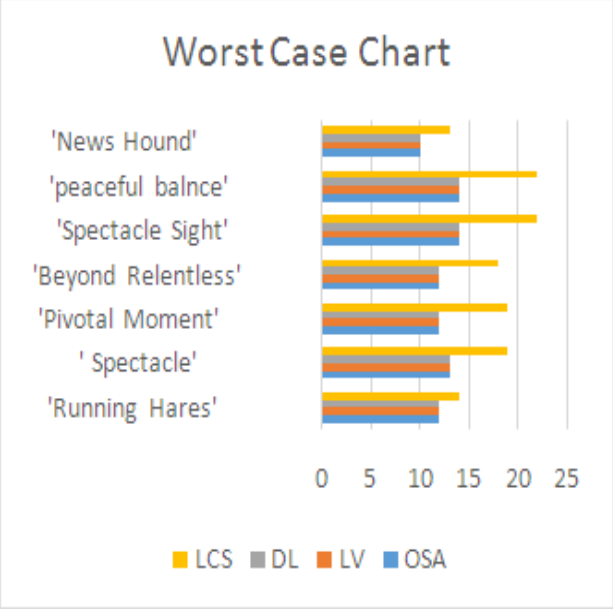

Figure II Difference chart of OSA, LV, DL, and LCSwith close to 0 means match

Now interestingly as shown in the above FigureI-IIchart the comparison with the string "Hound Hunting" showcase that all the best case strings and almost similar average words are matched what's more is, the worst case words are all discarded and got very high values. Also it's worth mentioning how typos on string "Hunting, Hound" where the string is having reversed word is not handled very well and got over 10 values i.e. they totally discarded it. And all algorithms in fig I show "Hunting, Hound" as a totally different word because of commabut wasable to handle the allworst cases in fig II. Moreover we can see that all the DL LV and OSA got similar values for all the best cases whereas the LCS performed just a little harsh in best case and average cases. While on the other hand comparing worst case strings as shown in Fig II they all have more than 10 values on all worst cases.However,we can clearly see a limitation that all these algorithm does not match string

when they are permuted with same number of characters as in the 'H.o.u.n.d. .H.u.n.t.i.n.g.' string case where it resulted 13 as totally different string. Overall they all yielded almost similar result. But let's check other algorithms result where we can also vary some values to get unique and better result. Shown in Figure IV chart take a look of how JW, Jaccard and cosine prompt for all identical rank and for q =1, 2, 3 simply varying only on the scaled distance rate. The calculations for which q=1 are clearly not interested in permuted character string. Jaro-Winkler again appears to think minimal about characters mixed, put arbitrarily or absent as long as the objective word's characters are available in exact order.

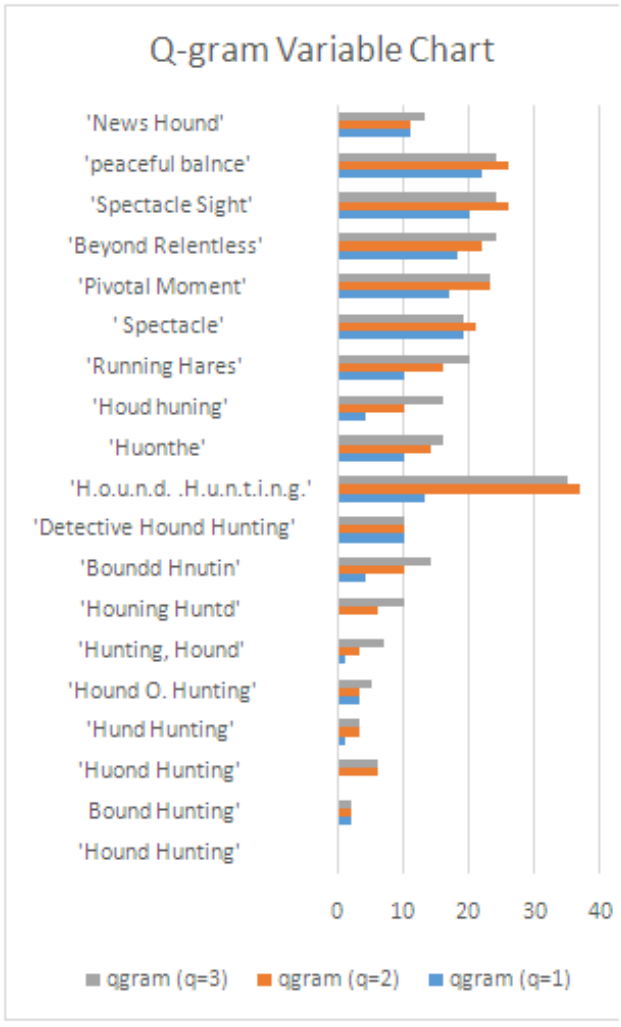

Figure IIIVariable Values Comparison of theQ-Gram with close to 0 means match

Which string matching technique to utilize relies upon the circumstances. On the off chance that we need to make up for grammatical errors then the varieties of the Levenshtein distance are of good utilization, also we can see cosine (q=2) shown in the figure IV and qgram (q=1) as shown in above graph in figureIII handled typos and misspelled kinds of mistakes very well. For example they were able to identify the similarity between reverse word string "Hunting, Hound". The metric could be enhanced by figuring the console design of input into the count. Let's say for English input console the separation amongst "test" and "rest" would then be minimal than the contrast amongst "test" and "best" as cleared by the layout of input .

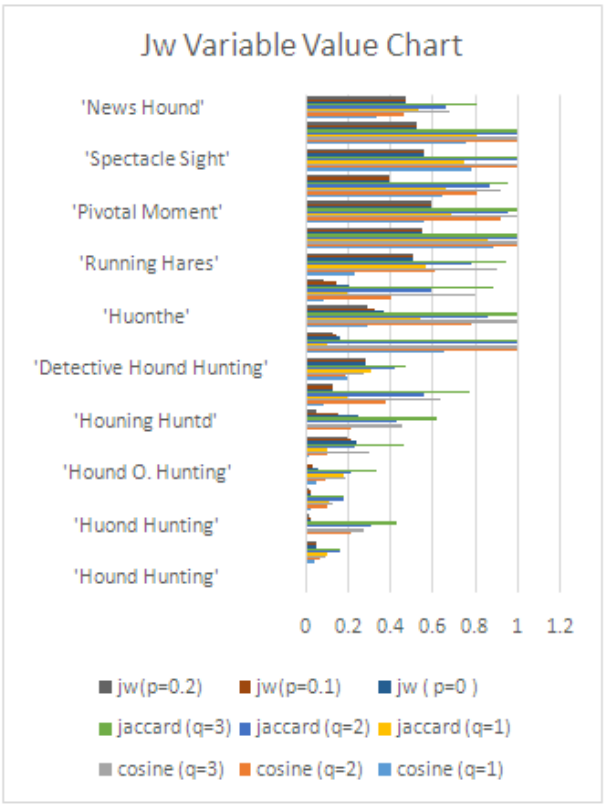

Figure IV Variable Values Comparison of the Jaccard, Cosine & JW with close to 0 means match

4. CONCLUSION

For future we would like to devise a more efficient and operation effective search technique on encrypted data using the string matching algorithms explained in this paper. We will devise a search technique which can handle typos query while being fast using hashing technique for reducing string matching operation on the index stored keyword for all encrypted files. We will also like explore the possibility of multidimensional search apart from basic typos search support; for enhancing user search experience.

REFERENCES

[1] E. Goh, "Secure Indexes," Cryptol. ePrint Arch. Rep. 2003/216, pp. 1–18, 2003.
[2] Y. Li, M. Li, and Y. Shen, "A Multi-attribute keyword Retrieval Mechanism for encrypted cloud data," Int. J. Secur. Its Appl., vol. 10, no. 12, pp. 335–346, 2016.
[3] M. Sharma, S. Yadav, M. Mathuria, and A. Tiwari, "Comparative Study on Search Techniques over Encrypted Data," in 2016 Online International Conference on Green Engineering and Technologies (IC-GET), 2016, pp. 1–6.
[4] Z. Fu, J. Xi, J. Wang, and X. Sun, "Document attribute-based keyword search over encrypted data," in 2014 10th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIH-MSP 2014, 2014, pp. 787–790.
[5] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling Personalized Search over Encrypted Outsourced Data with Efficiency Improvement," IEEE Trans. Parallel Distrib. Syst., vol. 27, no. 9, pp. 2546–2559, 2016.
[6] Z. Fu, F. Huang, X. Sun, A. Vasilakos, and C.-N. Yang, "Enabling Semantic Search based on Conceptual Graphs over

Encrypted Outsourced Data," IEEE Trans. Serv. Comput., 2016.

[7] Z. Fu, X. Sun, S. Ji, and G. Xie, "Towards efficient content-aware search over encrypted outsourced data in cloud," in IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications, 2016, vol. 2016–July, pp. 1–9.

[8] C. Hu and L. Han, "Efficient wildcard search over encrypted data," Int. J. Inf. Secur., vol. 15, no. 5, pp. 539–547, 2016.

[9] T. Suga, T. Nishide, and K. Sakurai, "Secure keyword search using bloom filter with specified character positions," in International Conference on Provable Security, 2012, vol. LNCS 7496, pp. 235–252.

[10] X. Zhu, Q. Liu, and G. Wang, "A Novel Verifiable and Dynamic Fuzzy Keyword Search Scheme over Encrypted Data in Cloud Computing," TrustCom/BigDataSE/ISPA, 2016.

[11] S. A. Mittal and R. C. Krishna, "Privacy Preserving Synonym Based Fuzzy Multi- Keyword Ranked Search Over Encrypted Cloud Data," in International Conference on Computing, Communication and Automation (ICCCA2016), 2016, pp. 1187–1194.

[12] Z. Fu, X. Wu, C. Guan, X. Sun, and K. Ren, "Toward Efficient Multi-Keyword Fuzzy Search over Encrypted Outsourced Data with Accuracy Improvement," IEEE Trans. Inf. Forensics Secur., vol. 11, no. 12, pp. 2706–2716, 2016.

[13] Y. Zhao, G. Wang, and Y. Yuan, "Fuzzy Keyword Search over Probabilistic XML Data," in 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), 2015, pp. 2523–2527.

[14] W. Ding, Y. Liu, and J. Zhang, "Chinese-keyword Fuzzy Search and Extraction over Encrypted Patent Documents," in 2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K), 2015, pp. 168–176.

[15] W. Jie, Y. Xiao, Z. Ming, and W. Yong, "A Novel Dynamic Ranked Fuzzy Keyword Search Over Cloud Encrypted Data," in 2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing, 2014, pp. 91–96.

[16] H. Tuo and M. Wenping, "An Effective Fuzzy keyword Search Scheme in Cloud Computing," in 2013 5th International Conference on Intelligent Networking and Collaborative Systems, 2013, pp. 786–789.

[17] Q. Xu, H. Shen, Y. Sang, and H. Tian, "Privacy-Preserving Ranked Fuzzy Keyword Search over Encrypted Cloud Data," in 2013 International Conference on Parallel and Distributed Computing, Applications and Technologies, 2013, pp. 239–245.

[18] L. Xue, R. Wuling, J. Guoxin, and Y. Jie, "A Solution Which Can Support Privacy Protection and Fuzzy Search Quickly under Cloud Computing Environment," in 2nd International Conference on Information Technology and Electronic Commerce（ICITEC 2014）, 2014, pp. 43–46.

[19] W. Song, B. Wang, Q. Wang, Z. Peng, W. Lou, and Y. Cui, "A privacy-preserved full text retrieval algorithm over encrypted data for cloud storage applications," J.Parallel Distrib. Comput., 2016.

[20] M. van der Loo, J. van der Laan, R Core Team, and N. Logan, "Package 'stringdist.'" pp. 1–24, 2018.

