A Novel Dynamic Secret Key Generation Protocol for Privacy Preserving Ranked Multi-Keyword Search

Dr.Ch.G.V.N.Prasad[1], R.Venkatesh[2], O.Mounika[3]
[1,2,3,4]*Department of Computer Science and Engineering,Sri Indu College of Engineering and Technology, Ibrahimpatnam, Telangana, India*

Abstract— Cloud computing has developed progressively prevalent for data owners to outsource their data to public cloud servers while consenting data users to reclaim this data. For isolation disquiets, a secure rifle over encrypted cloud data has stirred numerous research mechanisms underneath the particular owner model. Conversely, most cloud servers in practice do not just assist one owner, as an alternative, their sustenance gives multiple owners to share the assistances carried by cloud computing. In this proficient and confidentiality-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data, new schemes to deal with Privacy preserving Ranked Multi-keyword Search in a Multi-owner model (PRMSM) has been introduced. To facilitate cloud servers to execute secure search without knowing the actual data of both keywords and trapdoors, we thoroughly build a novel secure search protocol. To rank the search results and domain the privacy of relevance scores amongst keywords and files. To thwart the assailants from snooping secret keys and fantasizing to be legal data users submitting pursuits, a novel dynamic secret key generation protocol and a new data user authentication protocol is discussed.

Index Terms— Cloud computing, ranked keyword search, multiple owners, privacy preserving, dynamic secret key

1 INTRODUCTION

Computing is being transformed to a model consisting of services that are commoditized and delivered in a manner similar to utilities such as water, electricity, gas, and telephony. In such a model, users access services based on their requirements regardless of where the services are hosted. Several computing paradigms have promised to deliver this utility computing vision. Cloud computing is the most recent emerging paradigm promising to turn the vision of "computing utilities" into reality. A service offering computation resources is frequently referred to as Infrastructure as a Service (IaaS) and the applications as Software as a Service (SaaS)[1]. An environment used for construction, deployment, and management of applications is called PaaS (Platform as a Service).

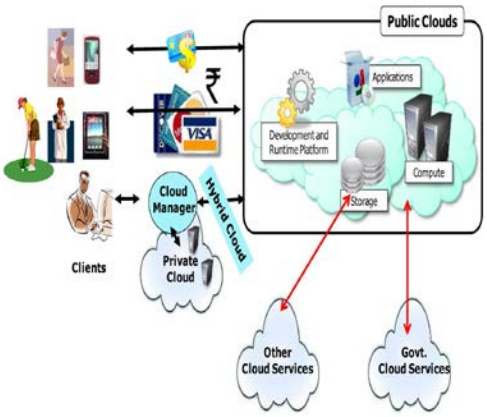


Fig.1: A bird's eye view of Cloud computing

Cloud computing delivers infrastructure, platform, and software (application) as services, which are made available as subscription-oriented services in a pay-as-you-go model to consumers. The price that CSPs (CloudService Providers) charge depends on the quality of service (QoS) expectations of CSCs (Cloud Service Consumers).Cloud computing fosters elasticity and seamless scalability of IT resources that are offered to end users as a servicethrough the Internet. Cloud computing can help enterprises improve the creation and delivery of IT solutions byproviding them with access to services in a cost-effective and flexible manner [2]. Clouds can be classified into three categories, depending on their accessibility restrictions and the deploymentmodel. They are:

- Public Cloud,
- Private Cloud, and
- Hybrid Cloud.

A public Cloud is made available in a pay-as-you-go manner to the general public users irrespective of their origin or affiliation. A private Cloud's usage is restricted to members, employees, and trusted partners of the organization. A hybrid Cloud enables the use of private and public Cloud in a seamless manner. Cloud computing applications span many domains, including business, technology, government, health care, smart grids, intelligent transportation networks, life sciences, disaster management, automation, data analytics, andconsumer and social networks. Various models for the creation, deployment, and delivery of these applications as Cloud services have emerged.

Cloud service providers (CSPs) would promise to certify owners' data security using purposes like virtualization and firewalls. Conversely, these mechanisms do not protect

owners' data privacy from the CSP itself, since the CSP holds full control of cloud hardware, software, and owners' data. Encryption on sensitive data formerly subcontracting can realmdata privacy beside CSP. Nevertheless, data encryption sorts the traditional data utilization service based on plaintext keyword search a very perplexing delinquent. A trifling solution to this problem is to move all the encrypted data and decrypt them nearby. Nonetheless, this method is evidently impracticable since it will cause a huge amount of communication overhead. Consequently, emerging a secure search service over encrypted cloud data is of overriding prominence. Secure search over encrypted data has recently attracted the interest of many researchers. Song et al. [3] first define and solve the problem of secure search over encrypted data. They propose the conception of searchable encryption, which is a cryptographic primitive that enables users to perform a keyword-based search on an encrypted dataset, just as on a plaintext dataset. Searchable encryption is additionally developed by [4], [6]. However, these schemes are concerned mostly with single or boolean keyword search. Encompassing these procedures for ranked multikeyword search will acquire heavy computation and storage costs. The main contributions of this proficient and confidentiality-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data are listed as Follows: (a) a multi-owner model for privacy preserving keyword search over encrypted cloud data is defined. (b) an efficient data user authentication protocol, which not only prevents attackers from eavesdropping secret keys and pretending to be illegal data users performing searches, but also enables data user authentication and revocation is defined. (c) a novel secure search protocol, which not only enables the cloud server to perform secure ranked keyword search without knowing the actual data of both keywords and trapdoors, but also allows data owners to encrypt keywords with self-chosen keys and allows authenticated data users to query without knowing these keys is systematically constructed.

2 RELATED WORK

2.1 Searchable Encryption

The earliest attempt of searchable encryption was made by Song et al. In [3], they propose to encrypt each word in a file independently and allow the server to find whether a single queried keyword is contained in the file without knowing the exact word. This proposal is more of theoretic interests because of high computational costs. Goh et al. propose building akeyword index for each file and using Bloom filter to accelerate the search [4]. Curtmola et al. propose building indices for each keyword, and use hash tables as an alternative approach to searchable encryption [5]. The first public key scheme for keyword search over encrypted data is presented in [6]. [7] and [8] further enrich the search functionalities of searchable encryption by proposing schemes for conjunctive keyword search. The searchable encryption cares mostly about single keyword search or boolean keyword search. Extending these techniques for ranked multi-keyword search will incur heavy computation and storage costs.

2.2 Secure Keyword Search in Cloud Computing

The privacy concerns in cloud computing motivate the study on secure keyword search. Wang et al. first defined and solved the secure ranked keyword search over encrypted cloud data. In [9] and [18], they proposed a scheme that returns the top-k relevant files upon a single keyword search. Cao et al. [10], [11], and Sun et al. [1], [12] extended the secure keyword search for multi-keyword queries. Their approaches vectorize the list of keywords and apply matrix multiplications to hide the actual keyword information from the cloud server, while still allowing the server to find out the top-k relevant data files. Xu et al. proposed MKQE (Multi-Keyword ranked Query on Encrypted data) that enables a dynamic keyword dictionary and avoids the ranking order being distorted by several high frequency keywords [13]. Li et al. [4], Chuah et al. [15], Xu et al. [16] and Wang et al. [7] proposed fuzzy keyword search over encrypted cloud data aiming at tolerance of both minor misprints and format inconsistencies for users' search input. [19] further proposed privacy-assured similarity search mechanisms over outsourced cloud data. In [10], a secure, efficient, and distributed keyword search protocol in the geo-distributed cloud environment. The system model of these previous works only consider one data owner, which implies that in their solutions, the data owner and data users can easily communicate and exchange secret information. When numerous data owners are involved in the system, secret information exchanging will cause considerable communication overhead. Sun et al. [2] and Zheng et al. [12] proposed secure attribute-based keyword search schemes in the challenging scenario where multiple owners are involved. However, applying CPABE in the cloud system would introduce problems for data user revocation, i.e., the cloud has to update the large amount of data stored on it for a data user revocation [14]. Additionally, they do not support privacy preserving ranked multi-keyword search. An proficient and confidentiality-Preserving Multi-

Keyword Ranked Search over Encrypted Cloud Data differs from previous studies regarding the emphasis of multiple data owners in the system model. An proficient and confidentiality-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data seeks a solution scheme to maximally relax the requirements for data owners and users, so that the scheme could be suitable for a large number of cloud computing users.

2.3 Order Preserving Encryption

The order preserving encryption is used to prevent the cloud server from knowing the exact relevance scores of keywords to a data file. The early work of Agrawal et al. proposed an Order Preserving symmetric Encryption (OPE) scheme where the numerical order of plain texts are preserved [13]. Boldyreva et al. further introduced a modular order preserving encryption in [4]. Yi et al [5] proposed an order preserving function to encode data in sensor networks. Popa et al. [6] recently proposed an ideal-secure order-preserving encryption scheme. Kerschbaum et al. [7] further proposed a scheme

which is not only idea-secure but is also an efficient order-preserving encryption scheme. However, these schemes are not additive order preserving. As a complementary work to the previous order preserving work, a new additive order and privacy preserving functions (AOPPF) are proposed. Data owners can freely choose any function from an AOPPF family to encode their relevance scores. The cloud server computes the sum of encoded relevance scores and ranks them based on the sum.

3 SYSTEM DESIGN

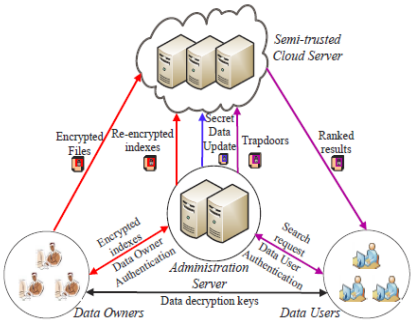


Fig. 1: Architecture of privacy preserving keyword search in a multi-owner and multi-user cloud model

3.1 Design Goals:

3.1.1 Ranked Multi-keyword Search over Multiowner:

The projected system should consent multi-keyword search over encrypted files which would be encrypted with dissimilar keys for altered data owners [10]. It also needs to allow the cloud server to rank the search results among unlike data owners and return the top-*k* results.

- **Data owner scalability:** The projected system should allow new data owners to enter this system without disturbing other data owners or data users, i.e., the scheme should support data owner scalability in a plug-and-play model.
- **Data user revocation:** The projected system should ensure that only legitimate data userscan perform correct rifles [9]. Moreover, once a data user is revoked, he can no longer perform accurate searches over the encrypted cloud data.
- **Security Goals:** The projected system should achieve the following security goals: 1) Keyword Semantic Security (Definition 1). We will prove that PRMSM achieves semantic security against the chosen keyword attack. 2) Keyword secrecy (Definition 2). Since the adversary A can know whether an encrypted keyword matches a trapdoor, we use the weaker security goal (i.e., secrecy), that is, we should ensure that the possibility for the adversary A to conclude the actual value of a keyword is insignificantly more than arbitrarily predicting [12]. 3) Relevance score secrecy. We should ensure that the cloud server cannot conclude the actual value of the encoded relevance scores.

3.2 Data User Authentication

To thwart attackers from pretending to be legal data users accomplishing searches and hurling statistical attacks based on the search result, data users must be authenticated before the administration server re-encrypts trapdoors for data users. Conventional authenticationmethods often follow [18] three steps. First, data requester and data authenticator share a secret key. Second, the requester encrypts his individually recognizable information and sends the encrypted data to the authenticator. Third, the authenticator decrypts the received data with and authenticates the decrypted data. Conversely, this method has two main drawbacks [17]. Since the secret key shared between the requester and the authenticator remains unaffected, it is easy to acquire repeat attack. Second, once the secret key is discovered to attackers, the authenticator cannot discriminate between the legal requester and the attackers[16]; the attackers can made-up to be legal requesters without being detected.

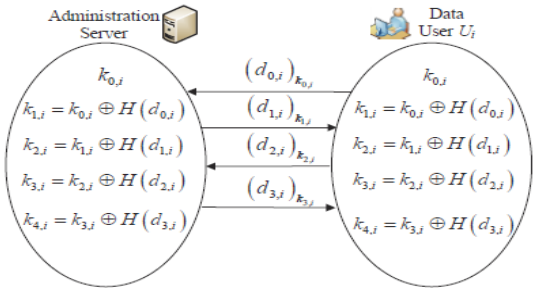


Fig.2: Example of data user authentication and dynamic Secret key generation

3.3 Data User Revocation

Dissimilar from previous works, data user revocation in this scheme does not need to re-encrypt and update large amounts of data stored on the cloud server. The administration server only needs to update the secret data stored on the cloud server. Accordingly, the earlier trapdoors will be perished [14]. Furthermore, without the help of the administration server, the repealed data user cannot produce the correct trapdoor. Hence, a data user cannot perform correct searches once he is revoked.

3.4 Keyword Encryption

For keyword encryption, the following conditions should be satisfied: first, distinct data owners use their own secret keys to encrypt keywords. Second, for the same keyword, it would be encrypted to distinct cipher-texts each time[15]. These belongings benefit the scheme for two reasons. First, losing the key of one data owner would not lead to the revelation of other owners' data[13]. Second, the cloud server cannot see any relationship among encrypted keywords.

3.5 Trapdoor Generation

To make the data users produce trapdoors securely, conveniently and efficiently, our projected system should mollify two main conditions. First, the data user does not need to ask a large amount of data owners for secret keys to engender trapdoors. Second, for the same keyword, the trapdoor generated each time should be distinct [12]. To meet this condition, the trapdoor generation is conducted in two steps: First, the data user produces trapdoors based on his search keyword and a random number. Second, the administration server re-

encrypts the trapdoors for the authenticated data user [19].

3.6 Keywords Matching among Distinct Data Owners

The cloud server stores all encrypted files and keywords of distinct data owners. The administration server will also store a secret data on the cloud server. Upon receiving a query request, the cloud will examine over the data of all these data owners[17]. The cloud processes the search request in two steps. First, the cloud contests the queried keywords from all keywords stored on it, and it gets a candidate file set. Second, the cloud ranks files in the candidate file set and finds the most top-*k* relevant files [18].

4 PROJECTED SYSTEM : PRIVACY PRESERVING RANKED SEARCH

The aforesaid section helps the cloud match the queried keywords, and acquire a candidate file set. Nonetheless, we cannot simply return non-distinct files to data users for the following two reasons. First, returning all candidate files would cause abundant communication overhead for the whole system. Second, data users would only apprehend the top-*k* relevantfiles corresponding to their queries [16]. We initially elucidate an order and privacy preserving encoding scheme. An additive order preserving and privacy preserving encoding scheme is demonstrated. The projected system to encode the [20] relevance scores and obtain the top-*k* search results is conferred.

4.1 Order and Privacy Preserving Function:

To rank the consequence score while preserving its privacy, the proposed function should satisfy the following conditions. 1) This function should preserve the order of data, as this helps the cloud server determine which file is more relevant to a certain keyword, according to the encoded relevance scores. 2) This function should not be revealed by the cloud server so that cloud server can make associations on encoded relevance scores without knowing their actual values. 3) Distinct data owners should have distinct functions such that enlightening the encoded value of a data owner would not lead to the leakage of encoded values of other data owners[19].

4.2 Ranking search results

In proficient and confidentiality-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data, the sum of the relevance scores as the metric to rank search results is used. The strategies of ranking search results based on the encoded relevance scores is introduced. First, the cloud computes the sum of encoded relevance scores between the file and matched keywords. Then the cloud ranks the sum of encoded relevance score with the following two conditions: (1) Two encoded data belong to the same data owner. Given that a data user issues a query and satisfies the[16] query. Then the cloud adds the encoded relevance score together and gets the relevance score.

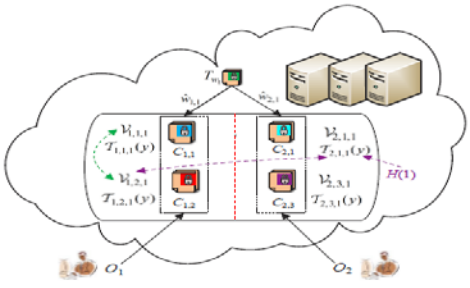
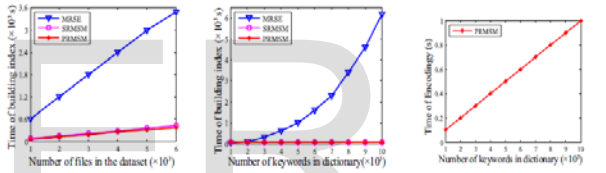


Fig. 3: Example of ranking search results

5 EXPERIMENTAL EVALUATION

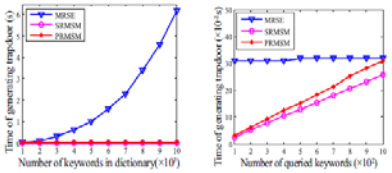
The efficiency of PRMSM is measured and compared it with its previous version, Secure Ranked Multi-keyword Search for Multiple data owners in cloud computing (SRMSM) [17], and the state of- the-art, privacy-preserving Multi-keyword Ranked Search over Encrypted cloud data (MRSE) [11], side by side. Since MRSE is only suitable for the single owner model, our PRMSM and SRMSM not only work well in multi-owner settings, but also outpace MRSE on many aspects.

The experiment programs are coded using the Python programming language on a PC with 2.2GHZ Intel Core CPU and 2GB memory. We implement all necessary routines for data owners to preprocess data files: [13],[10]for the data user to generate trapdoors, for the administrative server to re-encrypt keywords, trapdoors, and for the cloud server to perform ranked searches.



(a) For different size datasets with the same dictionary, u = 4000.
(b) For the same dataset with different dictionary sizes, n = 1000.
(c) Time cost of encoding relevance score (a subroutine of building index)

Fig. 6: Time cost of index construction.



(a) For different size of keyword dictionary with the same number of queried keywords, q = 100.
(b) For different number of queried keywords with the same size of keyword dictionary, u = 4000.

5.1 Index Construction

Fig. 6(a) shows that, given the same keyword dictionary (u=4000), time of index construction for these schemes escalate linearly with an increasing number of files, while SRMSM and PRMSM spend much less time on index construction. Fig. 6(b) reveals that, given the same number of files (n=1000), SRMSM and PRMSM ingest much less time than MRSE on constructing indexes. Furthermore, SRMSM and PRMSM are insensitive to the size of the keyword dictionary [9],[20]for index construction, while MRSE suffers a quadratic growth with the size of keyword dictionary increases. Fig. 6(c) shows the encoding efficiency of our proposed AOPPF. The time spent on encoding increases from 0.1s to 1s when the numberof keywords increases from 1000 to 10000. This time cost can be suitable.

5.2 Trapdoor Generation

Linked with index construction, trapdoor generation consumes relatively less time. Fig. 7(a) demonstrates that, given the same number of queried keywords (q=100), SRMSM and PRMSM are insensitive to the size of keyword dictionary on trapdoor generation and guzzles 0.026s and 0.031s, correspondingly. Temporarily, MRSE increases from 0.04s to 6.2s. Fig. 7(b) shows that, given the same number of dictionary size (u=4000), [17]when the number of queried keywords increases from 100 to 1000, the trapdoor generation time for MRSE is 0.31s, and remains unchanged. While SRMSM increases from 0.024s to 0.25s, PRMSM increases from 0.031s to 0.31s. We notice that PRMSM spends a little more time than SRMSM on trapdoor generation; the reason is that PRMSM familiarizes a further variable to ensure the randomness of trapdoors.

6. CONCLUSION

In proficient and confidentiality-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data, the tricky of secure multi-keyword search for multiple data owners and multiple data users in the cloud computing environment. Distinct from prior works, these schemes enable authenticated data users to achieve secure, expedient, and effectual searches over several data owners' data. To proficiently substantiate data users and distinguish attackers who steal the secret key and execute illegal searches, a novel dynamic secret key generation protocol and a innovative data user authentication protocol is discussed. To support the cloud server to accomplish secure search amid multiple owners' data encrypted with distinct secret keys, we thoroughly construct a novel secure search protocol. To rank the search results and preserve the privacy of relevance scores between keywords and files, we propose a novel Additive Order and Privacy Preserving Function family. Besides, it is shown that the slant is computationally effective, even for large data and keyword sets. The future work will consider the delinquent of secure fuzzy keyword search in a multi-owner paradigm and to implement the present scheme on the viable clouds.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Communication of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[2] C. Wang, S. S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacypreserving public auditing for secure cloud storage," *Computers, IEEE Transactions on*, vol. 62, no. 2, pp. 362–375, 2013.

[3] D.Song, D.Wagner, and A.Perrig, "Practical techniques for searches on encrypted data," in *Proc. IEEE International Symposium on Security and Privacy (S&P'00)*, Nagoya, Japan, Jan.2000, pp. 44–55.

[4] E. Goh. (2003) Secure indexes. [Online]. Available: http://eprint.iacr.org/

[5] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," in *Proc. ACM CCS'06*, VA, USA, Oct. 2006, pp. 79–88.

[6] D. B. et al., "Public key encryption with keyword search secure against keyword guessing attacks without random oracle," *EUROCRYPT*, vol. 43, pp. 506–522, 2004.

[7] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," in *Proc. Applied Cryptography and Network Security (ACNS'04)*, Yellow Mountain, China, Jun. 2004, pp. 31–45.

[8] L. Ballard, S. Kamara, and F. Monrose, "Achieving efficient conjunctive keyword searches over encrypted data," in *Proc. Information and Communications Security (ICICS'05)*, Beijing, China, Dec. 2005, pp. 414–426.

[9] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure ranked keyword search over encrypted cloud data," in *Proc. IEEE Distributed Computing Systems (ICDCS'10)*, Genoa, Italy, Jun. 2010, pp. 253–262.

[10] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy preserving multi-keyword ranked search over encrypted cloud data," in *Proc. IEEE INFOCOM'11*, Shanghai, China, Apr. 2011, pp. 829–837.

[11] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacypreserving multi-keyword ranked search over encrypted cloud data," *Parallel and Distributed Systems, IEEE Transactions on*,vol. 25, no. 1, pp. 222–233, 2014.

[12] W. Sun, B. Wang, N. Cao, M. Li, W. Lou, Y. T. Hou, and H. Li, "Verifiable privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 25, no. 11, pp. 3025–3035, 2014.

[13] Z. Xu, W. Kang, R. Li, K. Yow, and C. Xu, "Efficient multikeyword ranked query on encrypted data in the cloud," in *Proc. IEEE Parallel and Distributed Systems (ICPADS'12)*, Singapore, Dec. 2012, pp. 244–251.

[14] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in *Proc. IEEE INFOCOM'10*, San Diego, CA, Mar. 2010, pp. 1–5.

[15] M. Chuah and W. Hu, "Privacy-aware bedtree based solution for fuzzy multi-keyword search over encrypted data," in *Proc. IEEE 31th International Conference on Distributed Computing Systems (ICDCS'11)*, Minneapolis, MN, Jun. 2011, pp. 383–392.

[16] P. Xu, H. Jin, Q. Wu, and W. Wang, "Public-key encryption with fuzzy keyword search: A provably secure scheme under keyword guessing attack," *Computers, IEEE Transactions on*, vol. 62, no. 11, pp. 2266–2277, 2013.

[17] B. Wang, S. Yu, W. Lou, and Y. T. Hou, "Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud," in *IEEE INFOCOM*, Toronto, Canada, May 2014, pp. 2112–2120.

[18] C. Wang, N. Cao, K. Ren, and W. Lou, "Enabling secure and efficient ranked keyword search over outsourced cloud data," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 23, no. 8, pp. 1467–1479, 2012.

[19] C. Wang, K. Ren, S. Yu, and K. M. R. Urs, "Achieving usable and privacy-assured similarity search over outsourced cloud data," in *Proc. IEEE INFOCOM'12*, Or-

lando, FL, Mar. 2012, pp. 451–459.

[20] W. Zhang, Y. Lin, S. Xiao, Q. Liu, and T. Zhou, "Secure distributed keyword search in multiple clouds," in *Proc. IEEE/ACM IWQOS'14*, Hongkong, May 2014, pp. 370–379.