Semantic Searching Scheme over Encrypted Data by Applying Optimal Matching in Technique in Cloud

Uday Prasad #1, G Lakshmi Durga #2, U Thriveni #3, M Mounika #4, K Yamini #5

#1 Assistant Professor,Dept of CSE, Qis Institute of Technology,Ongole,Prakasam (Dt)
#2,3,4,5 B.Tech., Scholars, Dept of CSE, Qis Institute of Technology, Ongole, Prakasam (Dt)

Abstract

With the increasing adoption of cloud computing, a developing number of users re-appropriate their datasets to cloud. To protect the privacy, the datasets are usually encrypted prior to re-appropriating. Be that as it may, the normal practice of encryption makes the successful utilization of the data troublesome. For example, it is hard to search the given keywords in encrypted datasets. Many plans are proposed to make encrypted data searchable based on keywords. Be that as it may, watchword based search plans overlook the semantic representation information of users retrieval, and cannot totally meet with users search aim. Accordingly, how to plan a substance based search plan and make semantic search more successful and setting aware is a troublesome challenge. In this paper, we propose ECSED, a novel semantic search conspire based on the idea hierarchy and the semantic relationship between ideas in the encrypted datasets. ECSED utilizes two cloud servers. One is utilized to store the reevaluated datasets and return the ranked outcomes to data users. The other one is utilized to process the similarity scores between the records and the inquiry and send the scores to the primary server. To additionally improve the search proficiency, we use a tree-based file construction to organize all the record list vectors. We utilize the multi keyword ranked search over encrypted cloud data as our basic frame to propose two secure plans. The test results based on the real world datasets show that the plan is more proficient than past plans. We also demonstrate that our plans are secure under the known ciphertext model and the realized background model.

1 Introduction:

Cloud computing is another however gradual maturity model of big business IT infrastructure that gives high quality applications and administrations [1]. The cloud clients can rethink their local complex data system into the cloud to avoid the overhead of management and local storage. Be that as it may, the security of rethought data cannot be guaranteed, as the Cloud Service Provider (CSP) has entire control of the data. Thus, it is necessary to scramble data prior to rethinking them into cloud to secure the privacy of delicate data [13]. Li et al. [18] gave a safe privacy protecting re-appropriated classification in cloud computing. Notwithstanding, encryption for reevaluated data can ensure privacy against unauthorized behaviors, it also makes powerful data utilization, like search over

encrypted data, a very troublesome issue. As of late, many researchers have proposed an arrangement of productive search plots over encrypted cloud data. The general interaction of search plan can be separated into five steps: extracting report features, building a searchable file, generating search trapdoor, searching the record based on the trapdoor and returning the search results. These search plans give diverse inquiry capabilities, counting single watchword search [2, 3, 4, 5, 6], multi catchphrase search [7, 8, 9, 10], fluffy catchphrase search [9, 11] similarity search [12], and so on. Nonetheless, all the current searchable encryption plans, which think about keywords as the record feature, don't take the semantic relations between words into consideration, both in the means of extracting record features and generating search trapdoor. As we as a whole know, the semantic relations between words are various [14], like synonymy and domain correlation. Thinking about the potentially gigantic amount of reevaluated data archives in the cloud, the search accuracy and search proficiency are affected negatively if the semantic relations between words are not handled well. We currently give a detailed depiction of existing problems of the available searchable plans. Right off the bat, in the stage of extracting record features, the data owner registers the heaviness of each word in a record and then chooses t words with top-t loads as the feature of the record. In the interaction appeared above, each two words with various spelling are assumed uncorrelated, which is unreasonable. For example, two words "trousers", "pants" are extraordinary in the point of view of spelling, yet they are semantically similar. Clearly the heaviness of word is impacted in the event that semantic relations between words are disregarded and the accuracy of the archive features is affected thusly. Furthermore, during generating search trapdoor, the trapdoor is generated just based on the search keywords contribution by the data user, which is resolute, because it is unthinkable to expand the search keywords when the data user cannot express his search expectation well. In this case, a pointless archive can be returned for the data user or the really required records are not returned. Thus, it is important to understand the real search goal of the data user to avoid returning unnecessary records to improve search productivity, as the size of the archive set reevaluated into the cloud server is potentially tremendous. Thirdly, a search demand usually centers around a topic, and some search words can be viewed as the attribute of the topic, for example, birthday is an attribute of an individual. In existing search plans, an attribute value is usually treated as a catchphrase that disregards the relationship with the topic and results in larger catchphrase dictionary, and then negatively impacts the search accuracy and proficiency. Along these lines, it is a vital and challenging task to execute semantic search over encrypted data. In this paper, we propose a proficient searchable encrypted conspire based on idea hierarchy supporting semantic search with two cloud servers. An idea hierarchy tree is built based on domain ideas related information of the rethought dataset. We stretch out idea hierarchy to incorporate more semantic relations between ideas. With the help of broadened idea

hierarchy, archive features are extracted all the more absolutely and search terms are very much expanded based on the semantic relations between ideas. For each archive, two file vectors are generated, one is utilized to match ideas in the search demand and another one is used to decide if the value for an attribute is satisfied with the search demand. Correspondingly, the search trapdoor for a search demand also incorporates two vectors. The reason why we pick two cloud servers is that two servers can save a lot of time in search. One is utilized to figure the similarity between the reports vector and the trapdoors vector. Another one is utilized to rank outcomes and bring them back to users. Our commitments are summarized as follows:

1) We study the problem of the semantic search based on the idea hierarchy by utilizing two cloud servers. The idea hierarchy is reached out to store various semantic relations among ideas and used to broaden the search terms. To improve the productivity and security of the search, the retrieval interaction is parted into two free methodology.

2) We propose a technique to assemble the record file and search trapdoor based on the idea hierarchy to help semantic search, which channels reports by checking the attribute value and sorts related reports based on the quantity of matched search terms.

3) The security analysis indicates that our plan is secure in the threat models. A tree-based searchable record is developed to improve search proficiency. Examinations on real world datasets show that our plans are proficient.

2 Problem Formulation

2.1 System Model

Compared with the past version[6], we have an innovation on this adaptation is that we utilize two cloud servers to search, so we make another system model. There are four elements in our setting as demonstrated in Fig. 1: the data owner, the data user, the cloud server An and the cloud server B. Data owner: The data owner encodes the data held locally and uploads it to the cloud server. In this paper, an idea hierarchy is developed based on the domain ideas related information on the dataset and two record vectors for each archive of the dataset are generated based on the critical ideas of the record and the idea hierarchy. Then, at that point, the searchable list which is built with all the list vectors is shipped off the cloud A. Data users: The authorized data user makes a search demand. Then, at that point, the trapdoors which related to the keywords are generated. At last, the data user sends the trapdoors to the cloud B. Cloud Server A: The cloud server A has two capacities. One is putting away the re-appropriated dataset. The other one ranks the outcomes from the cloud B and returns the certain encrypted archives that satisfy the search basis to data users. Cloud Server B: The cloud server B is utilized to process the similarity scores between archives vector and trapdoors vector when it gets the trapdoor. After computing, the cloud B presents these outcomes to the cloud A.

2.2 Threat Model

The past paper[6] is basically presented the threat model. Our plan predominantly alludes to the double workers system [7] and the MRSE structure [7]. In this form, we believe the cloud worker to be semi-legitimate, which is embraced by most past works [7],[8],[9], in other words, who sincerely executes the convention as it is characterized and accurately returns the query items, however who may likewise attempt to surmise private data by dissecting the reevaluated dataset, accessible index and question assessment. What's more, we expect that there is no plot between two cloud workers. Because of what data the cloud workers learns, we study two threat model [15] as follows. Realized Ciphertext Model The known ciphertext model implies that the cloud workers can get to the scrambled data which contains the records and indexes reevaluated by data proprietors, however the workers can not comprehend the plaintext data in the lower layer of the ciphertext. Known Background Model In this more remarkable model, the cloud workers ought to have significantly more open data contrasted and known ciphertext model. These data incorporate the encoded data and the connection between given pursuit demands (trapdoors) and the data set about the factual data. As the example which can be assaulted in the present circumstance, the cloud workers can gather/perceive some recovered catchphrases by utilizing the known trapdoor data and the recurrence of records/watchwords.

2.3 Design Goal

We increment the piece of design goals to make this paper more clear than past paper[61]. To guarantee that our answers can be executed precisely and effectively under the previously mentioned threat

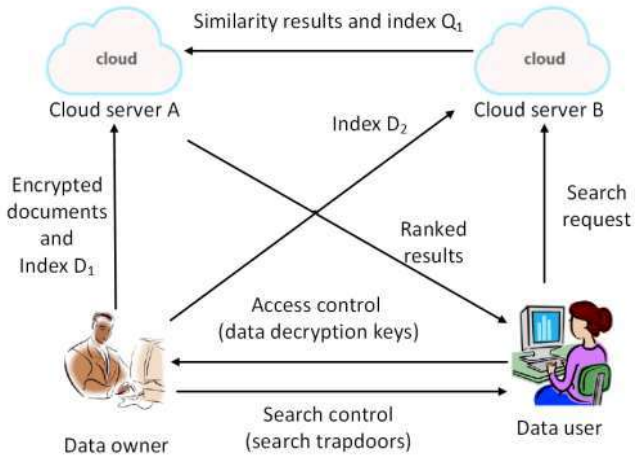

Fig. 1. System model

models, our plans should meet two prerequisites: semantic recovery dependent on concept hierarchy and privacy protecting. The semantic recovery dependent on concept hierarchy implies that our plan can figure the similarity scores between the data and the hunt solicitation and return the positioned results which fulfilled the inquiry solicitations of clients. In this section, we depict privacy protecting in detail. In the inquiry measures under the haze workers, our plans should meet the accompanying privacy assurance:

1) Data privacy. At the point when we give data records to clients, we additionally need to guarantee the privacy of the archive security, which is data privacy. To address this issue, the conventional symmetric cryptography has been proposed. The upside of this encryption is that we can utilize a symmetric key scrambled the data reports prior to outsouring.

2) Index privacy. Index privacy is that the cloud workers can not supposition the correspondence between the watchwords and the encoded reports through the scrambled index.
3) Concept privacy. In this paper, we accept that the concepts and the watchwords are connected to a certain degree. Thusly, we need to guarantee that the security trapdoor we created doesn't uncover the watchwords and the inquiry data of clients.
4) Trapdoor unlinkability. While the cloud workers recover archives, it can get to the created trapdoors. Consequently, we should ensure that the arbitrariness of trapdoor age. At the equivalent time, we need to guarantee that similar inquiries partner with many trapdoors. Along these lines, the cloud worker can not get connections which exist in these trapdoors.

3 Concept Hierarchy

A concept hierarchy is a coordinated concept set utilizing progressive strategy. In the hierarchy, the concepts at lower levels contain more explicit implications than those at higher levels. To make it straightforward, we give a model in Fig. 2 to show the concept hierarchy. As referenced previously, the concept hierarchy is developed dependent on the semantic connections between concepts, in other words, the concept hierarchy contains semantic data between concepts.

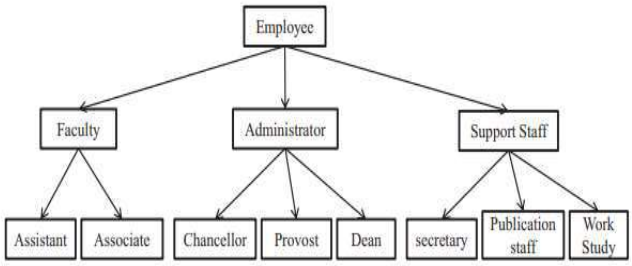

Fig. 2. A concept hierarchy for college employees.

3.1 Extended Concept Hierarchy

As we probably are aware, the content record is unstructured, yet the language arranging the content document exists semantic connection, which can be viewed as a certain construction. In this paper, we utilize an extended concept hierarchy to mean the semantic connection between concepts. The semantic relations contained in our extended concept hierarchy are appeared in shown. The principle distinction between our extended concept hierarchy and concept hierarchy is that, the concept can have trait, which can be doled out various qualities. A few concepts can have a few ascribes that are additionally concepts. We take concept "specialist" for instance. A "specialist" can have credits like "name", "sexual orientation, etc. The characteristic data can make the output more precise, as it makes the pursuit demand more explicit. Furthermore, as the concept what's more, trait are coordinated through the concept hierarchy, the semantic relationship is protected. We take Fig. 3 as an illustration to delineate the extended concept hierarchy. As appeared in Fig. 3, the trait "shading" of concept "shirt" has a esteem "red". Note that the concept went about as trait has not connection with different concepts. The hypernymy/hyponymy connection and

equivalent connection can likewise be found in Fig. 3. This passage shows that the way toward producing the extended concept hierarchy. From the start, we produce the concept hierarchy dependent on the area data of the re-appropriated dataset. And afterward we manage the dataset to broaden the concept hierarchy. The space concept hierarchy can be acquired by some current device, for example, WordNet [14], the hyponymy of which could be viewed as relationship of superclass and subclass, or some current concept hierarchy, for example, web catalog ODP(open registry project) [24]. As we as a whole know the substance of a report generally zero in regarding a matter that can be meant by concepts and relations of concepts. For instance, the sentence "A paper about economy is distributed in the paper on March 5, 2014" is the subject of an archive, which can be signified by concepts and relations among them. There are numerous subject extraction procedures with which the subjects of records are gotten. And afterward the concepts and its relations are created to expand the concept hierarchy. As WordNet incorporates every one of the semantic connections that embraced in the extended concept hierarchy, we use it to develop our extended concept hierarchy. The upsides of a specific characteristic concept are controlled by the reevaluated dataset.

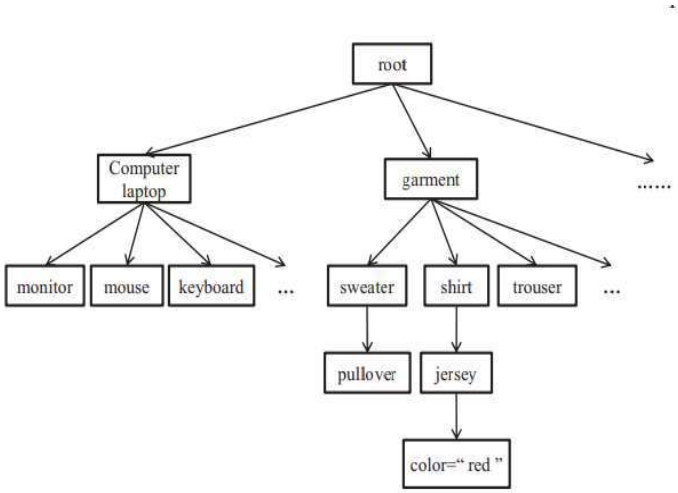

Fig. 3. An example of our extended concept hierarchy.

3.2 Semantic Similarity between Concepts

We presently give an itemized depiction of how to assess the semantic similarity between two concepts is given. The similarity between two concepts will be utilized in the phase of question to broaden search terms. After the concept hierarchy is assembled, the semantic similarity between two concepts can be determined. The similarity between two concepts is determined dependent on the distance of them in the concept hierarchy. Given two concepts c1 and c2, we mean the distance between them by dis(c1, c2). The similarity between them can be determined as sim(c1, c2) = 1 − dis(c1, c2). The issue of ascertaining the distance between two concepts has been concentrated by a few past work [3]. We make a few adjustments on their unique idea to accommodate our prerequisite. Every hub z in the concept hierarchy possesses a worth, signified by Score(z), which can be determined by Eqn. Score(z) = 1/2 k d(z) (3) where k is a factor which is characterized as 2 in this paper,

what's more, d(z) is the profundity of hub z in the concept hierarchy. Note that for the root hub of the concept hierarchy tree, we characterize d(root) = 0. For two concepts c1 and c2, we let concept cp be the nearest regular parent of them. Then, at that point, the distance between c1 and c2 can be acquired by Eqn.(2)

4 Secure Search Scheme

4.1 Generating Document Index Vector

As we introduce "attribute-value" relation in the hierarchy, two index vectors should be generated for each document in the dataset, one vector is used to match concepts in the search request and another one is used to determine whether the value for an attribute is satisfied with the search request. The process of generating these two n-dimension index vectors based on the extended concept hierarchy is shown as follows. For a document F, we denote its two index vectors by D1 and D2. Each dimension of D1, denoted by D1[i], corresponds to a node (stores concept ci) in the hierarchy. If F contains the concept ci , then D1[i] = 1, otherwise D1[i] = 0. Similarity, each dimension of D2, denoted by D2[i], corresponds to a node (stores concept ci) in the hierarchy.

4.2 Generating Trapdoor

A search request consists of several concepts. Once receiving a search request, the procedure calculates the semantic similarity between each search concept and its candidate concepts in the extended concept hierarchy using function defined in Section 3.1. For each search concept, the candidate concepts are its "brother", "father", and "direct child" nodes in the concept hierarchy. We let γ be a parameter to determine whether a candidate concept deserve to be added to search requests. To be specific, for a search concept c1 and its candidate concept c2, if $sim(c1, c2) \geq \gamma$, then we add c2 to the search request. Finally, the extended search request is generated. Note that we do not try to deal with attribute concepts in the concept extending process above. For a search request containing several concepts, two n-dimension vectors are also generated, one is used to store the information about concepts in the search request and another one is used to store the search restriction on attribute. For a search request Q, we denote its two search vectors by Q1 and Q2. The method of generating the value of each dimension of Q1 is similar to that for D1, that is, if Q contains ci , then Q1[i] = 1, otherwise Q1[i] = 0. For the extended search concepts, suppose cj is a concept extended from search concept ci and $sim(ci , cj) = \mu$, then Q1[j] = μ. For vector Q2, if there exist restriction on the value of attribute concept ci and the constraint value is λ, then Q2[i] = λ, otherwise Q2[i] = 0. We also take T in Fig. 4 as an example to illustrate the process. Assume that a search request Q contains concepts e, b, f, j after concept extending process, where j is a attribute concept whose value should satisfy val(j) > 1990. The search trapdoor Q1 and Q2 for Q. Given the index vectors of a document F and the search trapdoor of a search request Q, the search procedure is conducted as follows. Firstly, the procedure checks whether the document satisfies search restrictions included in search request using vectors D2 and Q2. Secondly, if D2

satisfies Q2, then the procedure computes D1 ·Q1 to obtain the similarity score of the document to the search request, where the value of D1 · Q1 indicates the number of matched concepts between F and Q. At last, all the related documents are sorted based on their similarity scores and the top-k related documents are returned to the user, where k is a parameter received from the user.

5 Related Work

Since Song et al. [1] proposed the notion of searching over encrypted cloud data, searchable encryption has received significant attention for its practicability in the past 20 years. Therefore, many works have made efforts on the security as well as functionality in the searchable encryption field. Along the research line about security, many works formulate the definitions of security as well as novel attack pattern against the existing schemes. Goh et al. [10] formulated a security model for document indexes known as semantic security against adaptive chosen keyword attack (IND-CKA), which requires the document indexes not to reveal contents of documents. However, we note that the definition of IND-CKA does not indicate that the queries must be secure.

Curtmola et al. [11] further improved security definitions for symmetric searchable encryption, then put forth chosen-keyword attacks and adaptive chosen-keyword attacks. Besides, Islam et al. [12] first introduced the access pattern disclosure used to learn sensitive information about the encrypted documents, then Liu et al. [13] presented a novel attack based on the search pattern leakage. Stefanov et al. [14] introduced the notions of forward security and backward security for the dynamic searchable encryption schemes that support data addition and deletion. Along another research line about functionality, many works introduced practical functions to meet the demand in practice, such as ranked search and semantic searching for improving search accuracy. Additionally, some works proposed verifiable searching schemes to verify the correctness of search results. Ranked Search over Encrypted Data. Ranked search means that the cloud server can calculate the relevance scores between the query and each document, then ranks the documents without leaking sensitive information. The notion of single keyword ranked search was proposed in [15] that used a modified one-to-many order-preserving encryption (OPE) to encrypt relevance scores and rank the encrypted documents. Cao et al. [16] first proposed a privacy-preserving multikeyword ranked search scheme (MRSE), which represents documents and queries with binary vectors and uses the secure kNN algorithm (SeckNN) [17] to encrypt the vectors, then use the inner product of the encrypted vectors as the similarity measure. Besides, Yu et al. [18] introduced homomorphic encryption to encrypt relevance scores and realize a multikeyword ranked search scheme under the vector space model. Recently, Kermanshahi et al. [19] used various homomorphic encryption techniques to propose a generic solution for supporting multi-keyword ranked searching schemes that can resist against several attacks brought by OPE-based schemes. Secure

Semantic Searching. A general limitation of traditional searchable encryption schemes is that they fail to utilize semantic information among words to evaluate the relevance between queries and documents.

Fu et al. [3] proposed the first synonym searchable encryption scheme under the vector space model to bridge the gap between semantically related words and given keywords. They first extended the keyword set from the synonym keyword thesaurus built on the New American Roget's College Thesaurus (NARCT), then used the extended keyword set to build secure indexes with SeckNN. Using the order-preserving encryption algorithm, [5] and [6] presented secure semantic searching schemes based on the mutual information model. Xia et al. [6] proposed a scheme that requires the cloud to constructs a semantic relationship library based on the mutual information used in [20]. However, any schemes based on the inverted index can calculate the mutual information model. Using the SeckNN algorithm, [7], [8], [2] proposed secure semantic searching schemes based on the concept hierarchy. For example, Fu et al. [8] proposed a central keyword semantic extension searching scheme which calculates weights of query words based on grammatical relations, then extends the central word based on the concept

hierarchy tree from WordNet. Inspired by word embedding used in plaintext information retrieval [21], [22], Liu et al. [9] introduced the Word2vec to represent both queries and documents as compact vectors.

6 Conclusion In this paper, to address the problem of semantic retrieval, we propose effective schemes based on concept hierarchy. Our solutions use two cloud servers for encrypted retrieval and make contributions both on search accuracy and efficiency. To improve accuracy, we extend the concept hierarchy to expand the search conditions. In addition, a tree-based index structure is constructed to organize all the document index vectors, which are built based on the concept hierarchy for the aspect of search efficiency. The security analysis shows that the proposed scheme is secure in the threat models. Experiments on real world dataset illustrate that our scheme is efficient.

References

1. L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds: towards a cloud definition," ACM SIGCOMM Comput. Commun. Rev., vol. 39, no. 1, pp. 50–55, 2009.
2. C. Wang, N. Cao, K. Ren, and W. Lou, "Enabling secure and efficient ranked keyword search over outsourced cloud data," IEEE TPDS, vol. 23, no. 8, pp. 1467–1479, 2012.
3. D. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Proc. of S&P, 2000.
4. R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," In Proc. of ACM CCS, 2006, pp. 79–88.

5. A. Swaminathan, Y. Mao, G.-M. Su, H. Gou, A. L.Varna, S. He, M. Wu, and D. W. Oard, "Confidentiality-preserving rank-ordered search," in Proc. of the 2007 ACM Workshop on Storage Security and Survivability, 2007, pp. 7–12.
6. S. Zerr, D. Olmedilla, W. Nejdl, and W. Siberski, "Zerber+r: Topk retrieval from a confidential index," in Proc. of EDBT, 2009, pp. 439–449
7. N. Cao, C. Wang, and M. Li, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," Parallel and Distributed Systems,IEEE Transactions on, vol.25,no.1, pp.222-233,2014.
8. W. Sun, B. Wang, N. Cao, M. Li, W. Lou, YT. Hou, and H.L, "Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking," in Proc. of ACM SIGSAC symposium on Information, computer and communications security, 2013, pp. 71–82.
9. M. Chuah and W. Hu, "Privacy-aware bedtree based solution for fuzzy multi-keyword search over encrypted data," in Proc. of the 31st ICDCSW, 2011, pp. 273–281.
10. Ayad Ibrahim, Hai Jin, Ali A. Yassin, and Deqing Zou, "Secure Rank-ordered Search of Multi-keyword Trapdoor over Encrypted Cloud Data," in Proc. of APSCC, 2012 IEEE Asia-Pacific, pp. 263–270.
11. J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in Proc. of IEEE INFOCOM'10 Mini-Conference, San Diego, CA, USA, March 2010, pp. 1–5.
12. C. Wang, K. Ren, S. Yu, K. Mahendra, and R. Urs, "Achieving Usable and Privacy-Assured Similarity Search over Outsourced Cloud Data," in Proc. of IEEE INFOCOM, 2012.
13. S. Kamara and K. Lauter, "Cryptographic cloud storage," in RLCPS, January 2010, LNCS. Springer, Heidelberg.
14. G. A. Miller, "WordNet: a lexical database for English," Communications of the ACM, vol.38, issue 11, pp. 39–41, 1995.
15. E.-J. Goh, "Secure indexes," Cryptology ePrint Archive, 2003, http://eprint.iacr.org/2003/216.
16. P. Scheuermann and M. Ouksel, "Multidimensional B-trees for associative searching in database systems," Information systems, vol. 7, issue 2, pp. 123–137, 1982.
17. D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in Proc. of TCC, 2007, pp. 535–554.
18. P. Golle, J. Staddon, and B. R. Waters, "Secure conjunctive keyword search over encrypted data," in Proc. of ACNS, 2004, pp. 31–45.
19. J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations,

and inner products," in Proc. of EUROCRYPT, 2008.
20. T. Okamoto and K. Takashima, "Adaptively Attribute-Hiding (Hierarchical) Inner Product Encryption," in Proc. of EUROCRYPT, 2012, pp. 591–608.
21. E. Shen, E. Shi, and B. Waters, "Predicate privacy in encryption systems," in Proc. of the 6th TCC, 2009.
22. V. Y. Lum and K. Meyer-Wegener, "An architecture for a multimedia database management system supporting content search," in Proc. of International Conference on Computing and Information, 1990, pp.304–313.
23. N. Guarino, C. Masolo, and G. Verete, "OntoSeek: Content-Based Access to the Web," IEEE Intelligent Systems, vol. 14, no. 3, pp. 70–80, 1999.
24. G. Varelas, E. Voutsakis, and P. Raftopoulou, "Semantic Similarity Methods in WordNet and their Application to Information Retrieval on the Web," in Proc. of 7th ACM international workshop on Web information and data management, 2005, pp. 10–16.
25. P. Cimiano, A. Pivk, L. Schmidt-Thieme, and S. Staab, "Learning taxonomic relations from heterogeneous sources," in Proc. of the ECAI 2004 Ontology Learning and Population Workshop, 2004.
26. W. Wang, W. Meng, and C. Yu, "Concept Hierarchy Based Text Database Categorization in a Metasearch Engine Environment," in Proc. of the First International Conference on Web Information Systems Engineering, 2000.
27. N. Nanas, V. Uren, and A. D. Roeck, "Building and applying a concept hierarchy representation of a user profile," in Proc. of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, 2003.