Verify & Decrypt multi-keyword ranked search over Encrypted data on Cloud server

[1]Ms. AartiBhirud, [2]Prof. Harish Barapatre

ABSTRACT

Cloud Computing is defined as a computing machine or group of machine which as a server/s, holds all the applications and document necessary all at one place so that any user can access it from anywhere whoever has a access to that server without actually using his/her physical machine.Any individual user who has permission to access the server can use the server'sprocessing power to run an application, store data, or perform any other computing task. Therefore, instead of using apersonal computer every-time to run the application, the individual can now run the application from anywhere in the world.For data privacy, data which is traverses using cloud computing need to be encrypted, which makes data utilization a very challenging task.In this paper, we evaluate and solve the challenging problem of privacy preserving multi-keyword ranked search over encrypted cloud data (MRSE). Among various multi-keyword techniques we choose the efficient similarity measure of "coordinate matching", i.e., as many matches as possible, to capture the relevance of data documents to the search query. We further use "inner product similarity" to quantitatively evaluate such similarity measure. We first propose a basic idea for the MRSE based on secure inner product computation, and then give two significantly improved MRSE schemes to achieve various stringent privacy requirements in two different threat models.

Keywords:- Cloud Computing, MRSE, Multi-keyword ranked Search, coordinate matching, inner product similarity, privacy preserving, Confidential Data.

1.INTRODUCTION

Cloud computing involves highly available massive compute and storage platforms offering a wide rangeof services. It enables convenient, on-demand network access to centralized resources that can be rapidly brings into effective action with great efficiency and minimum management overhead. The advantages of Cloud Computinginclude: On-demand self-service, easy network access, pooling, frequent resource elasticity, usage-based pricing. AsCloud Computing becomes predominant, more easily offended information is being centralized into the cloud. So, thefact that data owners and cloud server are not in the equal trusted domain may put the sourced data at risk. Thus, dataencryption makes effective data utilization a very challenging task given that there could be a large amount ofoutsourced data files. Cloud computing is called as a utility computing since it uses pay per use paradigm. Users have to pay for the usages. With the technology of cloud computing, users can access a variety of resources like programs, storage and application development platforms. Cloud is the extension of object oriented programming and it uses the concept of abstraction. Cloud computing is an emerging technology which helps as an utility, through which clients are going to store their data in the cloud server and using applications from a set of computing resources[1]. Here sensitive data is going to be centralized in the server. In some times the cloud server may leaks the data to hackers [2]. The data is going to encrypted before outsourced to achieve privacy. The encryption techniques increase the data utilization from a large amount of data. To retrieve data files we introduced keyword search mechanism. By this mechanism the users are going to retrieve the data files of their interest. In traditional search, encryption techniques the users are going to search data by using keywords without decrypting it, they support only Boolean keyword search only [2][10]. In cloud computing graded keyword search enhances the system usability by displaying the matching files by the help of relevance score. To achieve security and usability we introduce advanced cryptographic and information retrieval techniques, and using one-to-many order preserving symmetric encryption [3].

Basically there are three types of public cloud Services:

- Infrastructure as a service (IaaS): In this most basic cloud service model, IaaS providers offer computers, as physical or more often as virtual machines, and other resources.
- Platform as a service (PaaS): In the PaaS model, cloud providers deliver a computing platform typically including operating system, programming language execution environment, database, and web server. Application developers can develop and run their software solutions on a cloud platform without the cost and complexity of buying and managing the underlying hardware and software layers.
- Software as a service (SaaS): In the SaaS model, cloud providers install and operate application software in the cloud and cloud users access the software from cloud clients. The cloud users do not manage the cloud infrastructure and platform on which the application is running. This eliminates the need to install and run the application on the cloud user's own computers simplifying maintenance and support. What makes a cloud application different from other applications is its scalability.

2.DATA SECURITY ISSUES IN THE CLOUD

Privacy and Confidentiality

When the user host data to the cloud there should be some guarantee that the data will have limited authorized access. Unauthorized access to customer sensitive data by is another risk that can create potential threat to cloud data. Assurance should be provided with various security and privacy policies to secure cloud data.

Data Integrity

While preserving the privacy of the data, the cloud provider should make sure that the cloud data will persist the data integrity. Since while hosting data on cloud, it undergoes many changes and mechanisms, so cloud provider should ensure the data integrity. For compliance purposes, it may be necessary to have exact records as to what data was placed in a public cloud, when it occurred, what virtual memories (VMs) and storage it resided on, and where it was processed.

Data location and Relocation

Cloud Computing offers a high degree of data mobility. Consumers do not always know the location of their data. However, when an enterprise has some sensitive data that is kept on a storage device in the Cloud, they may want to know the location of it. They may also wish to specify a preferred location (e.g. data to be kept in India). This, then, requires a contractual agreement, between the Cloud provider and the consumer that data should stay in a particular location or reside on a given known server. Also, cloud providers should take responsibility to ensure the security of systems (including data) and provide robust authentication to safeguard customers' information. Another issue is the movement of data from one location to another. Data is initially stored at an appropriate location decide by the Cloud provider. However, it is often moved from one place to another. Cloud providers have contracts with each other and they use each-othersresources.

Data Availability

Customer data is normally stored in chunk on different servers often residing in different locations or in different Clouds. In this case, data availability becomes a major legitimate issue as the availability of uninterruptible and seamless provision becomes relatively difficult.

Storage, Backup and Recovery

When you decide to move your data to the cloud the cloud provider should ensure adequate data resilience storage systems. At a minimum they should be able to provide RAID (Redundant Array of Independent Disks) storage systemsalthough most cloud providers will store the data in multiple copies across many independent servers. To meet the effective data retrieval need, the large amount of documents demand the cloud server to perform result relevance ranking, instead of returning undifferentiated results. Such ranked search system enables data users to find the most relevant information quickly, rather than burdensomely sorting through every match in the content collection [3]. Ranked search can also elegantly eliminate unnecessary network traffic by sending back only the most relevant data, which is highly desirable in the "pay-as-you-use" cloud paradigm. For privacy protection, such ranking operation, however, should not leak any keyword related information. On the other hand, to improve the search result accuracy as well as to enhance the user searching experience, it is also necessary for such ranking system to support multiple keywords search, as single keyword search often yields far too coarse results. As a common practice indicated by today's web search engines (e.g., Google search), data users may tend to provide a set of keywords instead of only one as the indicatorof their search interest to retrieve the most relevant data. And each keyword in the search request is able to help narrow down the search result further. "Coordinate matching" [4], i.e., as many matches as possible, is an efficient similaritymeasure among such multi-keyword semantics to refine the result relevance, and has been widely used in the plaintext information retrieval (IR) community. However, how to apply it in the encrypted cloud data search system remains a very challenging task because of inherent security and privacy obstacles, including various strict requirements like the data privacy, the index privacy, the keyword privacy, and many others. In the literature, searchable encryption [5]–[13] is a helpful technique that treats encrypted data as documents and allows a user to securely search through a single keyword and retrieve documents of interest. However, direct application of these approaches to the secure large scale cloud data utilization system would not be necessarily suitable, as they are developed as crypto primitives and cannot accommodate such high service-level requirements like system usability, user searching experience, and easy information discovery. Although some recent designs have been proposed to support Boolean keyword search [14]–[21] as an attempt to enrich the search flexibility, they are still not adequate to provide users with acceptable result ranking functionality. Our early work [22] has been aware of this problem, and provided a solution to the secure ranked search over encrypted data problem but only for queries consisting of a single keyword. How to design an efficient encrypted data search mechanism that supports multi-keyword semantics without privacy breaches still remains a challenging open problem. In this paper, for the first time, we define and solve the problem of multi-keyword ranked search over encrypted clouddata (MRSE) while preserving strict system-wise privacy in the cloud computing paradigm. Among various multi-keyword semantics, we choose the efficient similarity measure of "coordinate matching", i.e., as many matches as possible, to capture the relevance of data documents to the search query. Specifically, we use "inner product similarity" [4], i.e.the number of query keywords appearing in a document, to quantitatively evaluate such similarity measure of that document to the search query. During the index construction, each document is associated with a binary vector as a sub-index where each bit represents whether corresponding keyword is contained in the document. The search query is also described as a binary vector where each bit means whether corresponding keyword appears in this search request, so the similarity could be exactly measured by the inner product of the query vector with the data vector. However, directly outsourcing the data vector or the query vector will violate the index privacy or the search privacy. To meet the challenge of supporting such multi-keyword semantic without privacy breaches, we propose a basic idea for the MRSE using secure inner product computation, which is adapted from a secure k-nearest neighbor (kNN) technique [23], and then give two significantly improved MRSE schemes in a step-by-step manner to achieve various stringent privacy requirements in two threat models with increased attack capabilities. Our contributions are summarized as follows,

1. For the first time, we explore the problem of multi-keywordranked search over encrypted cloud data, andestablish a set of strict privacy requirements for such a secure cloud data utilization system.
2. We propose two MRSE schemes based on the similaritymeasure of “coordinate matching” while meeting differentprivacy requirements in two different threat models.
3. Thorough analysis investigating privacy and efficiencyguarantees of the proposed schemes is given, and experimentson the real-world dataset further show theproposed schemes indeed introduce low overhead oncomputation and communication.

The remainder of this paper is organized as follows. InSection II, we introduce the system model. Section III describes the MRSE framework and privacy requirements, followed by section IV, which describes the proposed schemes, and conclude the paper in Section V.

3.SYSTEM MODEL

Considering a cloud data hosting service involving three different entities, as illustrated in Fig. 1: the data owner, the data user, and the cloud server. The data owner has a collection of data documents F to be outsourced to the cloud server in the encrypted form C. To enable the searching capability over C for effective data utilization, the data owner, before outsourcing, will first build an encrypted searchable index I from F, and then outsource both the index I and the encrypted document collection C to the cloud server.

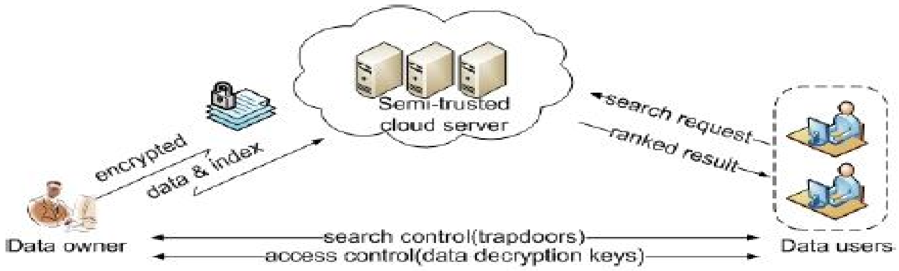


Fig 1: Architecture of the search over encrypted cloud data

In the architecture we have three entities:

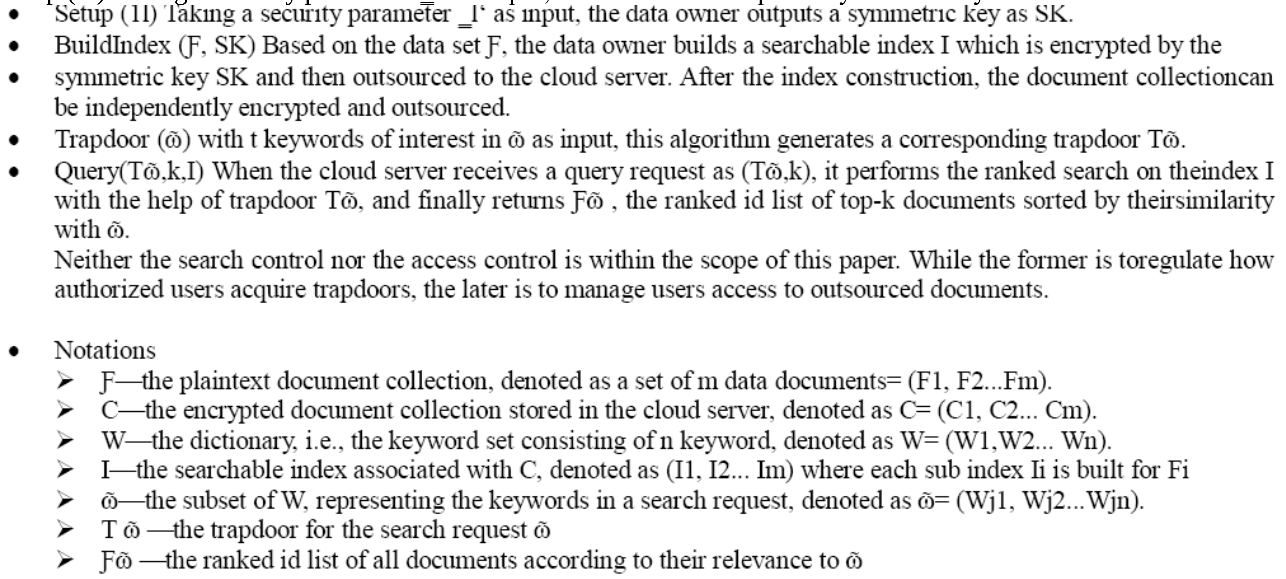
1. Data owner: Data xection of data files that he wants to outsource into the cloud server in encrypted format, this will increase effective data utilization.
2. Data user: When the data user wants to search the required files he enters a keyword in a secret form.
3. Cloud server: It is the place where a pool of data files and different applications can store.

Previously user can selects the files in the form of a plain text files. This is ailing under access the files. There is no perfect decryption technique to access the files of representation process. Here we introduce encryption based secure keyword searching mechanism. It can provide efficient solution for accessing the data. It is a good usability to display the effective matching details files [4]. These matching files are extracted with relevance score. This kind of matching files are retrieved with efficient mechanism. It can provide the results with guaranteed mechanism. All the files are collected with encryption format. All encrypted files are given weight in implementation process. These kinds of approaches show the better result in implementation. The search result is displayed according to relevance score which improves file retrieval accuracy .In information retrieval process we maintain an inverted index to represent file ID’s and relevance scores.

3.MRSE FRAMEWORK

In this section, we define the framework of multi-keyword ranked search over encrypted cloud data (MRSE) and establish various strict systemize privacy requirements for such a secure cloud data utilization System. The operations on the data documents are not shown in the framework since the data owner could easily employ the traditional symmetric key cryptography to encrypt and then outsource data. With focus on the index and query, the MRSE system

consists of four algorithms as follows--
Setup (1l) Taking a security parameter _l‘ as input, the data owner outputs a symmetric key as SK.



- Setup (1l) Taking a security parameter _l‘ as input, the data owner outputs a symmetric key as SK.
- BuildIndex (Ƒ, SK) Based on the data set Ƒ, the data owner builds a searchable index I which is encrypted by the
- symmetric key SK and then outsourced to the cloud server. After the index construction, the document collectioncan be independently encrypted and outsourced.
- Trapdoor (ω̃) with t keywords of interest in ω̃ as input, this algorithm generates a corresponding trapdoor Tω̃.
- Query(Tω̃,k,I) When the cloud server receives a query request as (Tω̃,k), it performs the ranked search on theindex I with the help of trapdoor Tω̃, and finally returns Ƒω̃ , the ranked id list of top-k documents sorted by theirsimilarity with ω̃.

  Neither the search control nor the access control is within the scope of this paper. While the former is toregulate how authorized users acquire trapdoors, the later is to manage users access to outsourced documents.

- Notations
  - Ƒ—the plaintext document collection, denoted as a set of m data documents= (F1, F2...Fm).
  - C—the encrypted document collection stored in the cloud server, denoted as C= (C1, C2... Cm).
  - W—the dictionary, i.e., the keyword set consisting of n keyword, denoted as W= (W1,W2... Wn).
  - I—the searchable index associated with C, denoted as (I1, I2... Im) where each sub index Ii is built for Fi
  - ω̃—the subset of W, representing the keywords in a search request, denoted as ω̃= (Wj1, Wj2...Wjn).
  - T ω̃ —the trapdoor for the search request ω̃
  - Ƒω̃ —the ranked id list of all documents according to their relevance to ω̃

PRIVACY REQUIREMENTS FOR MRSE

With general privacy description, we explore and establish a set of strict privacy requirements specifically for the MRSE framework. As for the data privacy, the data owner can resort to the traditional symmetric key cryptography to encrypt the data before outsourcing, and successfully prevent the cloud server from prying into the outsourced data. With respect to the index privacy, if the cloud server deduces any association between keywords and encrypted documents from index, it may learn the major subject of a document, even the content of a short document [26]. Therefore, the searchable index should be constructed to prevent the cloud server from performing such kind of association attack. While data and index privacy guarantees are demanded by default in the related literature, various search privacy requirements involved in the query procedure are more complex and difficult to tackle as follows.

Keyword Privacy: As users usually prefer to keep their search from being exposed to others like the cloud server, the most important concern is to hide what they are searching, i.e., the keywords indicated by the corresponding trapdoor. Although the trapdoor can be generated in a cryptographic way to protect the query keywords, the cloud server could do some statistical analysis over the search result to make an estimate. As a kind of statistical information, document frequency (i.e., the number of documents containing the keyword) is sufficient to identify the keyword with high probability [27]. When the cloud server knows some background information of the dataset, this keyword specific information may be utilized to reverse-engineer the keyword.

Trapdoor Un-linkability: The trapdoor generation function should be a randomized one instead of being deterministic. In particular, the cloud server should not be able to deduce the relationship of any given trapdoors, e.g., to determine whether the two trapdoors are formed by the same search request. Otherwise, the deterministic trapdoor generation would give the cloud server advantage to accumulate frequencies of different search requests regarding different keyword(s), which may further violate the aforementioned keyword privacy requirement. So the fundamental protection for trapdoor un-linkability is to introduce sufficient non-determinacy into the trapdoor generation procedure.

Access Pattern: Within the ranked search, the access pattern is the sequence of search results where every search result is a set of documents with rank order. Specifically, the search result for the query keyword set $\widetilde{W}$ is denoted as $F_{\widetilde{W}}$, consisting of the id list of all documents ranked by their relevance to$\widetilde{W}$.Although a few searchable encryption works, e.g., [17] has been proposed to utilize private information retrieval (PIR) technique [28], to hide the access pattern, our proposed schemes are not designed to protect the access pattern for the efficiency concerns. This is because any PIR based technique must “touch” the whole dataset outsourced on the server which is inefficient in the large scale cloud system.

4.MRSE I SCHEME

In our more advanced design, instead of simply removing the extended dimension in the query vector as we plan to do at the first glance, we preserve this dimension extending operation but assign a new random number t to the extended dimension in each query vector. Such a newly added randomness is expected to increase the difficulty for the cloud server to learn the relationship among the received trapdoors. In addition, as mentioned in the keyword privacy requirement, randomness should also be carefully calibrated in the search result to obfuscate the document frequency and diminish the chances for re-identification of keywords. Introducing some randomness in the final similarity score is an effective way towards what we expect here. More specifically, unlike the randomness involved in the query vector, we insert a dummy keyword into each data vector and assign a random value to it. Each individual vector Di is extended to (n+2)-dimension instead of (n + 1), where a random variable εirepresenting the dummy keyword is stored in the extended dimension. The whole scheme to achieve ranked search with multiple keywords over encrypted data is as follows.

- Setup The data owner randomly generates a (n + 2)-bit vector as S and two (n+2)×(n+2) invertible matrices {M1,M2}. The secret key SK is in the form of a 3-tuple as {S,M1,M2}.
- BuildIndex(F, SK) The data owner generates a binary data vector Di for every document Fi, where each binary bit Di[j] represents whether the corresponding keyword Wjappears in the document Fi. Subsequently, every plaintext sub-index $\overline{D_i}$is generated by applying dimension extending and splitting procedures on Di. These procedures are similar with those in the secure kNNcomputation except that the (n + 1)-th entry in $\overline{D_i}$isset to a random number εi, and the (n + 2)-th entry in is set to 1 during the dimension extending.$\overline{D_i}$is therefore equal to (Di, εi, 1).
- Trapdoor($\widetilde{W}$) With t keywords of interest in fW as input,one binary vector Q is generated where each bit Q[j]indicates whether Wj∈$\widetilde{W}$is true or false. Q is firstextended to n + 1-dimension which is set to 1, andthen scaled by a random number r □= 0, and finallyextended to a (n + 2)-dimension vector as $\bar{Q}$where thelast dimension is set to another random number t. $\bar{Q}$ is therefore equal to (rQ, r, t). After applying the samesplitting and encrypting processes as above, the trapdoor
- Query($T_{\widetilde{W}}$, k, I) With the trapdoor $T_{\widetilde{W}}$, the cloud servercomputes the similarity scores of each document Fi. WLOG, we assume r > 0. After sorting allscores, the cloud server returns the top-k ranked id list$F_{\widetilde{W}}$.

5.CONCLUSION

In this paper, for the first time we define and solve the problem of multi-keyword ranked search over encrypted cloud data, and establish a variety of privacy requirements. Among various multi-keyword semantics, we choose the efficient similarity measure of “coordinate matching”, i.e., as many matches as possible, to effectively capture the relevance of outsourced documents to the query keywords, and use “inner product similarity” to quantitatively evaluate such similarity measure. For meeting the challenge of supporting multi-keyword semantic without privacy breaches, we propose a basic idea of MRSE using secure inner product computation. Then we give two improved MRSE schemes to achieve various stringent privacy requirements in two different threat models. Thorough analysis investigating privacy and efficiency guarantees of proposedschemes is given, and experiments on the real world dataset show our proposed schemes introduce low overhead on both computation and communication.

REFERENCES

[1] Ning Cao, Cong Wang, Ming Li, KuiRen, and Wenjing Lou, “Privacy-Preserving Multi-keyword Ranked Search over Encrypted Cloud Data” IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS VOL:25 NO:1 YEAR 2014

[2] 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.

[3] S. Kamara and K. Lauter, “Cryptographic cloud storage,” in RLCPS, January 2010, LNCS. Springer, Heidelberg.

[4] A. Singhal, “Modern information retrieval: A brief overview,” IEEE Data Engineering Bulletin, vol. 24, no. 4, pp. 35–43, 2001.

[5] I. H. Witten, A. Moffat, and T. C. Bell, “Managing gigabytes: Compressing and indexing documents and images,” Morgan Kaufmann Publishing, San Francisco, May 1999.

[6] D. Song, D. Wagner, and A. Perrig, “Practical techniques for searches on encrypted data,” in Proc. of S&P, 2000.

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

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

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

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

[11] M. Bellare, A. Boldyreva, and A. ONeill, “Deterministic and efficiently searchable encryption,” in Proc. of CRYPTO, 2007.

[12] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi, “Searchable encryption revisited: Consistency properties, relation to anonymous ibe, and extensions,” J. Cryptol., vol. 21, no. 3, pp. 350–391, 2008.

[13]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 IEEEINFOCOM’10 Mini-Conference, San Diego, CA, USA, March 2010.

[14] D. Boneh, E. Kushilevitz, R. Ostrovsky, and W. E. S. III, “Public key encryption that allows pir queries,” in Proc. of CRYPTO, 2007.

[15] P. Golle, J. Staddon, and B. Waters, “Secure conjunctive keyword search over encrypted data,” in Proc. of ACNS, 2004, pp. 31–45.

[16] L. Ballard, S. Kamara, and F. Monrose, “Achieving efficient conjunctive keyword searches over encrypted data,” in Proc. of ICICS, 2005.

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

[18] R. Brinkman, “Searching in encrypted data,” in University of Twente, PhD thesis, 2007.

[19] Y. Hwang and P. Lee, “Public key encryption with conjunctive keyword search and its extension to a multi-user system,” in Pairing, 2007.

[20] J. Katz, A. Sahai, and B. Waters, “Predicate encryption supportingdisjunctions, polynomial equations, and inner products,” in Proc. OfEUROCRYPT, 2008.

[21] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, “Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption,” in Proc. of EUROCRYPT, 2010. [21] E. Shen, E. Shi, and B. Waters, “Predicate privacy in encryption systems,” in Proc. of TCC, 2009.

[22] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, “Secure ranked keyword search over encrypted cloud data,” in Proc. of ICDCS’10, 2010.

[23] W. K. Wong, D. W. Cheung, B. Kao, and N. Mamoulis, “Secure knn computation on encrypted databases,” in Proceedings of the 35thSIGMOD international conference on Management of data, 2009, pp. 139–152.

[24] S. Yu, C. Wang, K. Ren, and W. Lou, “Achieving secure, scalable, and fine-grained data access control in cloud computing,” in Proc. OfINFOCOM, 2010.

[25] C. Wang, Q. Wang, K. Ren, and W. Lou, “Privacy-preserving public auditing for data storage security in cloud computing,” in Proc. OfINFOCOM, 2010.

[26] S. Zerr, E. Demidova, D. Olmedilla, W. Nejdl, M. Winslett, and S. Mitra, “Zerber: r-confidential indexing for distributed documents,” in Proc. OfEDBT, 2008, pp. 287–298.

[27] S. Zerr, D. Olmedilla, W. Nejdl, and W. Siberski, “Zerber+r: Top-k retrieval from a confidential index,” in Proc. of EDBT, 2009, pp. 439–449.

[28] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai, “Cryptography from anonymity,” in Proceedings of the 47th Annual IEEE Symposiumon Foundations of Computer Science, 2006, pp. 239–248.

[29] W. W. Cohen, “Enron email dataset,” http://www.cs.cmu.edu/□enron/