Toward Privacy-Assured Cloud Data Services with Flexible Search Functionalities

Ming Li
Dept. of CS, Utah State University
Email: ming.li@usu.edu

Shucheng Yu
Dept. of CS, University of Arkansas
Email: sxyu1@ualr.edu

Wenjing Lou and Y. Thomas Hou
Virginia Tech
Email: {wjlou,thou}@vt.edu

***Abstract*—User privacy has been a major concern against the widespread adoption of the cloud technology. A full-fledged cloud data service should effectively support data utilization tasks, especially flexible data search functionalities, while simultaneously achieve user privacy assurance and meet practical system-level performance requirements. In this position paper, we identify the importance and challenges of designing privacy-assured, *flexible* and *practically efficient* search mechanisms for outsourced cloud data services. In particular, we focus on two representative types of flexible search functionalities: ranked keyword search, and search over structured data. Although these functionalities are already prevalent in information retrieval in the plaintext domain, realizing them in the encrypted domain requires non-trivial effort and is relatively new. In light of this, we first describe several existing technical approaches proposed by us and other researchers, and identify their advantages and limitations. We also discuss the open research directions and provide some possible ideas for further investigation. We believe the presented results will inspire more research towards making privacy-assured search in the cloud practical and useful.**

I. Introduction

The cloud promises to provide massively scalable data storage and computation services to the society at a reduced cost, due to the centralized management of elastic resources [2]. In this emerging computing platform, the cloud provider, application developers, and end-users can all reap benefits. Especially, the end-users can outsource large volumes of data and workloads to the cloud and enjoy the virtually unlimited computing resources in a pay-per-use manner. Indeed, many companies, organizations and individual users have already adopted the cloud platform to facilitate their business operation, research, or everyday needs by tapping in the cloud's shared pool of configurable computing resources [1].

Yet, despite the tremendous business and technical advantages, privacy concern is one of the primary hurdles that prevent the widespread adoption of the cloud, especially because a large amount of users' outsourced data involve sensitive personal information. Well-known examples include financial and medical records, and social network profiles. In current practices, by outsourcing data to the cloud the end-users must also yield exclusive control of their data and computations. It is difficult for users to fully trust the cloud, due to its non-bug-free deployment, low degree of transparency, and reported incidents of unauthorized access from either inside or outside of the cloud [24].

One promising approach for users to take back control of their data is to encrypt them beforehand. However, while data encryption guarantees data confidentiality, it rules out many normal computations on the data available in the plaintext domain. Especially, a basic need is *search* service, which can quickly sort out relevant information from huge amounts of data. For example, for enterprise/organizational end-users, database search is an everyday operation that underlies their corporate business intelligence [11]. More advanced functions like data mining also uses statistical query as a key primitive. Individual cloud users such as mobile subscribers would like convenient and intelligent services that help them with daily activity planning, which heavily involves query and answer. Therefore, it is highly desirable to enable *privacy-assured search services* over encrypted outsourced data, which ideally does not leak any sensitive user information to the cloud, such as business secrets or private personal activities. Without being able to effectively utilize the outsourced data, the cloud will merely be a remote storage with limited values.

However, to design a practical privacy-assured search mechanism remains a very challenging task due to a number of requirements that shall be met simultaneously. First, such a mechanism should be practically feasible in terms of computational complexity. Otherwise, outsourced data services would be too costly for the users while the cloud's economic value will diminish. Second, the data search mechanisms must support a wide range of functionalities with high usability in order to be useful in reality. They should support rich query semantics, versatile query predicates, and various forms of data structure. Third, they need to achieve sound security guarantees without introducing other restricting system assumptions. They should strike good balances between security guarantees and practical performance, in comparison to state-of-the-art techniques. Unfortunately, it is generally known that the more complex the function being evaluated on the ciphertext, the harder to guarantee a high level of security.

Theoretically, query computations over encrypted data can be accomplished by general secure computation techniques [30], which has been extensively studied in both cryptography and theoretical computer science communities. Recently there have been notable advances in secure computation outsourcing [14], [13], [23] using fully-homomorphic encryption (FHE) [14]. However, to date these techniques still incur excessively high computation and/or communication complexity, and applying them to our daily data utilization needs would be far from practical [10]. On the other hand, "searchable encryption" (SE) techniques have been proposed, which are specific solutions addressing the needs for private search over encrypted data. Yet, most existing SE schemes can only deal with Boolean keyword searches, where queries are expressed by Boolean formulas and encrypted documents that satisfy the formula are returned. Such search functionalities are still quite basic, and are unlikely to have wide scale applications alone.

In this paper, we posit that *flexibility and efficiency* are two key factors in order to make privacy-assured search techniques practical for cloud computing. We identify two types of flexible search functions with practical importance: ranked search, and search over structured data. On the one hand, the ranked search semantic greatly enhances the relevance of returned search results and reduces communication overhead, which is highly desirable for building usable cloud data services. On the other hand, a large portion of today's online data is represented using rich structures beyond simple text-form. Without being able to utilize those structured data, the economic potential of cloud services will not be fully realized. Thus, we survey recent research advances in these two topics, and give insights on possible approaches to overcome the limitations of current symmetric searchable encryption techniques, to enable more flexible privacy-assured search.

The rest of this paper is organized as follows. In Sec. II, we present the general problem setting of a privacy-assured cloud storage system with search capabilities. In Sec. III, we briefly review the existing Boolean keyword searchable encryption schemes. In Sec. IV, we describe several existing technical approaches for flexible privacy-assured search that are proposed by us and other researchers, along with our discussions on future directions. Sec. V concludes the paper.

II. System Architecture, Service Envision and Requirements

A. System Architecture

We describe a general data service outsourcing architecture involving three (types of) entities (Fig. 1). The *data owner* (or data contributor) is one or multiple entities who generate, encrypt data and upload them to the cloud server. The owner can either be an organization or an individual. The *cloud server* within a cloud service provider (CSP) possess significant storage and computation resources and provides them to the end-users in a pay-per-user manner. There are one or more *data users* in the system, who may need to perform queries over the outsourced data in order to extract useful information. To enable search access by the users, the data owner usually generates and distributes cryptographic keys or "trapdoors" to the users, either actively or upon users' requests. When a user wants to initiate a query, he/she submits a corresponding trapdoor to the server, who carries out the search and returns the results in an encrypted format. In some situations, the data user and data owner can be the same physical entity.

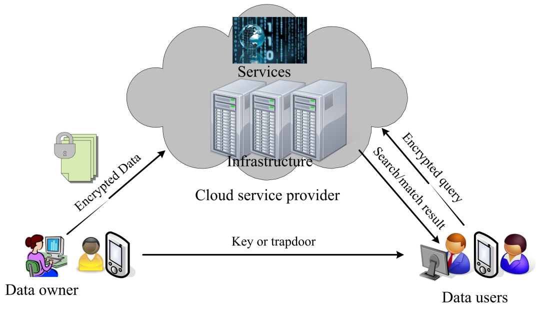

Fig. 1. System Architecture for Outsourced Cloud Data Services

B. Service Envision

The above system architecture captures a wide range of cloud data services, and the application scenario varies depending on the types of data owners and users. The data owner and user can be the same person; for example, Alice uploads her personal albums to Dropbox and wants to search for a particular photo afterwards. Or if we consider corporate data owner, a company may outsource its business records to a cloud server to enjoy the low-cost storage. At the same time, its employee in the auditing department may need to search the business database for the records containing sensitive activities. Alternatively, the data owner may be an individual while the user can be a company. For instance, consider a pervasive healthcare application where each patient uploads her health monitoring data periodically to a third-party medical service. The latter operates in the cloud, and will provide health reports to the patient by evaluating patient's data using some flexible diagnostic criteria.

C. Threat Models

A typical assumption is that the cloud servers are semi-trusted. That means they try to find out as much private information in the stored data as possible, but will honestly follow the protocol in general. This assumption is in line with the current technology trend and business model. The CSP and the data owners are not in the same trust domain, and there may exist curious employees inside the CSP who access user data for their own benefits. In addition, some users may also try to access/utilize the data beyond their privileges, either individually or in collusion with each other.

Depending on the information available to the adversary, a basic threat model (known-ciphertext) is to assume that the cloud server only possesses the encrypted data and index. In a stronger known-background model, the cloud server may possess some statistical information about the outsourced dataset, for example, the distributions of term frequency and document frequency [31].

D. Privacy-Assured Search Framework and Design Space

We first give a general algorithmic framework corresponding to the searchable cloud data service envisioned above.

- Setup$(1^\ell) \rightarrow \{K\}$*This algorithm takes a security parameter ℓ as input and outputs a secret key K which* *is later used by the owner or distributed to authorized users.*

- $\mathsf{IndexGen}(\mathcal{F}, K) \rightarrow \{\mathcal{F}_K, \mathcal{I}_K\}$ *This algorithm builds an encrypted index $\mathcal{I}$ corresponding to dataset $\mathcal{F}$ using the secret key K, and then outsource $\{\mathcal{F}_K, \mathcal{I}_K\}$ to the cloud server. The input data $\mathcal{F}$ can be encrypted using a separate algorithm.*
- $\mathsf{TrapdoorGen}(K, \mathcal{T}) \rightarrow \{\mathcal{T}_K\}$ *This algorithm encrypts the search function $\mathcal{T}$ using secret key K, outputs a trapdoor $\mathcal{T}_K$.*
- $\mathsf{Search}(\mathcal{T}_K, \mathcal{I}_K) \rightarrow \{\mathcal{R}_K\}$ *This algorithm evaluates the function $\mathcal{T}$ (e.g., keyword search or similarity matching) on the encrypted data index, obtains the evaluation result $\mathcal{T}(\mathcal{I})$ on the data index (e.g., 0 or 1), based on which it returns a subset of encrypted data (i.e., $\mathcal{R}_K$).*

Both the owner's outsourced data and user's queries on those data may contain sensitive information and need protection against the adversary. More specifically, the system should meet the following security requirements:

- *Data & index privacy*: Without the secret key K, no one, including the cloud server, should be able to learn sensitive information from the owner's private data. Similarly, they should not be able to deduce sensitive information underlying the data index, because the index is often associated with data itself.
- *Search Privacy*: Users' most important concern is to hide the search criteria they are evaluating on the data, e.g., their query keywords. These should not be derivable from the encrypted trapdoor and data/index sent to the cloud server, even when the server possesses some additional background information such as keyword distribution. In addition, queries shall have *unlinkability*, i.e., the cloud server shall not learn whether they have the same criteria.

Efficiency. A privacy-assured data search scheme should have low computation, communication and storage overheads. For such a scheme to be deployed in a large-scale cloud storage system with economically practicality, we argue that the search process should be completed within both constant communication round, and computation time (independent of the database size). In general, the privacy guarantee conflicts with efficiency. For example, it is more private to prevent the cloud server from learning the access pattern, i.e., the sequence of returned data. However, current techniques that protect it such as oblivious RAM [16] and private information retrieval (PIR) [18] are still far from practical. The former requires logarithmic-rounds, while the latter must "touch" the whole dataset outsourced to the server. Thus we do not discuss them in this paper, while those studies are of independent interest.

III. Background on Searchable Encryption

Early searchable encryption schemes were based on Boolean keyword queries. Song et. al's seminal work on SE enables in-line equality keyword search within a text document [25], while the computation complexity is linear with the document's size. Later, many researchers developed SE schemes that allow searches over encrypted keyword indexes, either based on public key cryptography (PKC) [5], [20] or symmetric key cryptography (SKC) [15], [17], [12]. In general, existing PKC-based schemes allow for more expressive queries than SKC-based ones, and enable data contributions from multiple owners/contributors. However, they are often less efficient due to the use of bilinear pairing operations. Thus, there has been a significant interest in developing efficient SKC-based SE mechanisms. Curtmola et. al. were the first to propose a symmetric SE scheme (SSE) with security guarantees under rigorous definitions [12]. Their scheme only supports single-keyword queries which is restrictive for practical use. Most of the following up works [3], [19] deal with Boolean keyword search or combinations of them such as conjunctive formula. However, we note that there has been a great interest to develop privacy-assured search mechanisms that support more expressive and usable query functionalities beyond Boolean search. For example, Li et al. proposed to enhance the search usability by allowing fuzzy keyword matches that tolerate user input errors [21], [27]. It is a recent and ongoing effort to investigate flexible search schemes over encrypted data.

IV. Flexible Privacy-Assured Search Schemes

A. Secure Ranked Search over Encrypted Data

An especially important functionality in plaintext information retrieval, is to support ranking mechanisms over search results according to user-specified relevance criteria. In addition, it is very common in that users evaluate Boolean formulas consisting of multiple keywords on a large dataset to further limit their interest. The same needs to be achieved in the encrypted domain.

To this end, Wang et. al. proposed a ranked symmetric searchable encryption (RSSE) scheme [26]. Their idea is to adopt concepts from the information retrieval (IR) community for result ranking, while exploiting recent advances in symmetric key cryptography to ensure user privacy. In particular, to support result ranking, a ranking function is used to calculate relevance scores of files to a given search request:

$$Score(t, F_d) = \frac{1}{|F_d|} \cdot (1 + \ln f_{d,t}), \qquad (1)$$

where t is the search keyword (term), $f_{d,t}$ denotes the term frequency (TF) of keyword t in file F_d, and $|F_d|$ is the number of indexed keywords in F_d.

To achieve privacy assurance, a straightforward yet ideally secure RSSE scheme can be derived based on the existing SSE solution in [12], but requires two rounds of interactions between the user and the cloud server which incurs high communication overhead. Thus, they adopt a recent cryptographic primitive – *order preserving symmetric encryption* (OPSE) [4] to obtain practical performance, where in OPSE the numerical ordering of the plaintexts is preserved after encryption. High efficiency is maintained by slightly weakening the security guarantee to be "as-strong-as-possible". Specifically, during the search operation the relevance order (OPSE encrypted relevance scores) of each document is revealed to the server.

In this way, efficient relevance score ranking can be done as in the plaintext domain. However, because the original OPSE is a deterministic encryption scheme, this still leaks much information. If the server has some background information on the dataset, such as the distribution of relevance scores for each plaintext keyword, it could reverse-engineer the keyword.

To break this determinacy, the authors propose one-to-many order-preserving mapping (OPM) which maps the same relevance score to different encrypted values. They incorporate the unique file IDs together with the plaintext as the random seed in the final ciphertext chosen process in OPSE. Thus, the same plaintext will no longer be deterministically assigned to the same ciphertext, but instead a random value within the randomly assigned bucket in a range $\mathcal{R}$. Furthermore, they use different keys to encrypt the relevance score for different posting lists (documents containing each keyword) to make the OPM more indistinguishable.

The RSSE scheme achieves *data and index privacy*, because the relevance scores in the searchable index are encrypted using OPSE with OPM. The highly flattened one-to-many mapping and the fully randomized score-to-bucket assignment in OPSE makes it difficult for the adversary to predict the original plaintext score distribution by observing the ciphertext. In addition, this scheme hides the search keyword from the adversary. But since the trapdoor is deterministic, it does not provide unlinkability. For efficiency, the encrypted index generation and search operations can both be finished within seconds for 1000 documents.

The above method cannot directly handle multiple-keyword ranked search, because the order of OPSE's ciphertext will not be preserved for the sum of multiple relevance scores. To support secure multi-keyword ranked search over encrypted data (MRSE), in [7] we proposed to adopt another similarity measure from IR community, "*coordinate matching*", which captures the relevance of documents to a query through the number of query keywords appearing in a document. Each document index and the query is described as a binary vector, respectively, such that the similarity is measured by the dot product of the two vectors. In order to protect the index privacy and the search privacy, we shall encrypt the index and query vectors, and compute the similarity score over ciphertexts.

To this end, we propose a secure inner-product computation mechanism that adapts ideas from the secure k-nearest neighbor (kNN) scheme in [28]. Basically, the search operation should compute the dot product between a query vector $\vec{q}$ and each data (index) vector $\vec{p_i}$. However, a straightforward application of the scheme in [28] is not secure as it linearly preserves the dot product, by which the server can statistically analyze similarity scores for two queries differing in one keyword to learn that keyword (called "scale analysis attack"), especially in the known-background model. Therefore, to build a secure MRSE that preserves search privacy, we obfuscate the document frequency to diminish the chances for re-identification of keywords. In particular, we propose to add randomness to both the data vector and query vector in order to blind the exact similarity score from the server. The randomness is added on-the-fly, by extending both vectors with dummy random keywords.

In our MRSE scheme, the data and index privacy are achieved since the encryption algorithm is secure in the known-ciphertext model. In addition, under the known-background model *search privacy* is achieved, as well as trapdoor unlinkability. It introduces nearly constant search overhead with the increase of keywords; in contrast, in other multiple-keyword search schemes [17], [6] this is linear.

B. Privacy-Assured Searches over Structured Data

Large portions of online data are not stored in a simple text form; rather they are have rich data structures. For example, graphs has been increasingly used to model complicated and schemaless data, such as social network graphs, medical workflows, relational databases, chemical compounds, and personal images. As more and more sensitive structured data are outsourced, users need to effectively search them even when they are encrypted. Recently, Chase and Kamara proposed structured encryption [9] to handle private access to parts of a large graph in encrypted form; yet only simple operations such as neighbor queries are supported.

In [8], Cao et. al. proposed a privacy-preserving graph containment query scheme (PPGQ). In the plaintext domain, graph containment means checking subgraph isomorphism. As direct checking is NP-complete, the principle of "filtering-and-verification" is usually used, where a feature-based index is pre-built for each graph. When data graphs are stored in encrypted form in the cloud, the filtering method based on plaintext index is no longer available. Thus, to support privacy-assured graph search over encrypted data, they convert both the data and query graphs into binary vectors, and use their dot product as the filtering condition. Each bit within a data/query graph's vector represents whether the corresponding feature is subgraph-isomorphic to that graph or not; only when the dot product of the two vectors equals to the number of query features, the matching data graph will be returned.

The PPGQ scheme is again based on the kNN technique in [28]. As a straightforward application of it would violate the query privacy, the authors propose another randomization technique. Essentially, only the dot products between a query graph and the matching graphs will be preserved, while all of the others are randomized. The scheme achieves both data and index privacy, and *search privacy* under the known-background model. Performance evaluation shows the TrapdoorGen and Search functions are very efficient.

C. Discussions and Future Directions

The works mentioned above have a common characteristic: they relax the privacy guarantees (namely, "as-strong-as-possible") to achieve higher efficiency performance. While there exists formal privacy definitions for searchable encryption that reveals the access pattern [12], for "as-strong-as-possible" schemes, how to formally analyze the privacy level given various known background information remains an interesting yet important open problem. Addressing it may require tools from information theory and statistics.

There are many interesting research issues worth further investigating. For multi-keyword ranked search, it is desirable to enable advanced relevance criteria such like the ones commonly used in IR. The problem is how to hide the sum of multiple keywords' relevance scores from the server, who may possess statistical distribution information to re-identify the search keywords. A possible approach would be to use a similar procedure as in MRSE to build randomized document indexes and query vectors. For search on structured data, the PPGQ scheme does not handle graphs with labeled nodes, which, however is quite common in practice as graph nodes usually have concrete and different meanings. On the other hand, because of noises that are usually contained in graph databases, exact graph containment queries could often return very few results [29]. Thus, enabling similarity searches over encrypted graphs is another important functionality for outsourced graph-structured data. It is worthwhile to explore a variety of similarity measures used in the plaintext graphical-IR domain. In addition, for wider applicability in different scenarios, can we make public-key based searchable encryption more practical and secure? We studied privacy enhancements of public-key based multi-dimensional queries in [22], while its efficiency is still well behind symmetric-key based solutions. Finally, it is also interesting to ask if more complex data utilization functions can be efficiently evaluated on encrypted data. For example, running database join/merge queries, or graph algorithms on structured data.

V. Concluding Remarks

In this position paper, we identify the problem and challenges of enabling privacy-assured flexible search functionalities for cloud data services. Recent research advances in this field are surveyed, which suggest that achieving semantically-rich, usable and efficient search on encrypted data is possible without sacrificing much privacy guarantee. The steady evolution of this field will need to bring expertise from cryptography, database and information retrieval communities.

Acknowledgements. The authors would like to thank the anonymous reviewers for their helpful comments. W. Lou is partly supported by NSF under Grant CNS-1155988, and Y. T. Hou is partly supported by NSF under Grant CNS-0910531.

References

[1] Salesforce. http://www.salesforce.com/platform/cloud-platform/.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. Above the clouds: A berkeley view of cloud computing, Feb 2009.

[3] M. Bellare, A. Boldyreva, and A. O'Neill. Deterministic and efficiently searchable encryption. In *Proceedings of Crypto'07, volume 4622 of LNCS*. Springer, 2007.

[4] A. Boldyreva, N. Chenette, Y. Lee, and A. O'Neill. Order-preserving symmetric encryption. In *Proceedings of Eurocrypt'09, volume 5479 of LNCS*. Springer, 2009.

[5] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In *Proc. of EUROCRYP'04*, 2004.

[6] D. Boneh and B. Waters. Conjunctive, subset, and range queries on encrypted data. In *Proc. of TCC'07*, pages 535–554, 2007.

[7] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou. Privacy-preserving multi-keyword ranked search over encrypted cloud data. In *INFOCOM, 2011 Proceedings IEEE*, pages 829–837. IEEE, 2011.

[8] N. Cao, Z. Yang, C. Wang, K. Ren, and W. Lou. Privacy-preserving query over encrypted graph-structured data in cloud computing. In *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*, pages 393–402. IEEE, 2011.

[9] M. Chase and S. Kamara. Structured encryption and controlled disclosure. *Advances in Cryptology-ASIACRYPT 2010*, pages 577–594, 2010.

[10] Y. Chen and R. Sion. On securing untrusted clouds with cryptography. In *Proceedings of the 9th annual ACM workshop on Privacy in the electronic society*, pages 109–114. ACM, 2010.

[11] R. Chow, P. Golle, M. Jakobsson, E. Shi, J. Staddon, R. Masuoka, and J. Molina. Controlling data in the cloud: outsourcing computation without outsourcing control. In *Proceedings of the 2009 ACM workshop on Cloud computing security*, pages 85–90. ACM, 2009.

[12] R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky. Searchable symmetric encryption: improved definitions and efficient constructions. In *Proc. of ACM CCS'06*, 2006.

[13] R. Gennaro, C. Gentry, and B. Parno. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. *Advances in Cryptology–CRYPTO 2010*, pages 465–482, 2010.

[14] C. Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 169–178. ACM, 2009.

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

[16] O. Goldreich and R. Ostrovsky. Software protection and simulation on oblivious rams. *Journal of the ACM (JACM)*, 43(3):431–473, 1996.

[17] P. Golle, J. Staddon, and B. Waters. Secure conjunctive keyword search over encrypted data. In *ACNS 04: 2nd International Conference on Applied Cryptography and Network Security*, pages 31–45. Springer-Verlag, 2004.

[18] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai. Cryptography from anonymity. In *Proc. of FOCS*, pages 239–248, 2006.

[19] S. Kamara and K. Lauter. Cryptographic cloud storage. In *Proceedings of Financial Cryptography: Workshop on Real-Life Cryptographic Protocols and Standardization 2010*, January 2010.

[20] J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *EUROCRYPT'08*, pages 146–162, Berlin, Heidelberg, 2008. Springer-Verlag.

[21] 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.

[22] M. Li, S. Yu, N. Cao, and W. Lou. Authorized private keyword search over encrypted data in cloud computing. In *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*, pages 383–392. IEEE, 2011.

[23] M. Naehrig, K. Lauter, and V. Vaikuntanathan. Can homomorphic encryption be practical? In *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*, pages 113–124. ACM, 2011.

[24] S. Pearson and A. Benameur. Privacy, security and trust issues arising from cloud computing. In *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, pages 693 –702, 30 2010-dec. 3 2010.

[25] D. Song, D. Wagner, and A. Perrig. Practical techniques for searches on encrypted data. In *Proc. of IEEE S & P '00*, 2000.

[26] 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.

[27] C. Wang, Q. Wang, and K. Ren. Towards secure and effective utilization over encrypted cloud data. In *Distributed Computing Systems Workshops (ICDCSW), 2011 31st International Conference on*, pages 282–286. IEEE, 2011.

[28] W. K. Wong, D. W. Cheung, B. Kao, and N. Mamoulis. Secure knn computation on encrypted databases. In *Proc. of SIGMOD*, 2009.

[29] X. Yan, F. Zhu, P. Yu, and J. Han. Feature-based similarity search in graph structures. *ACM Transactions on Database Systems (TODS)*, 31(4):1418–1453, 2006.

[30] A. Yao. Protocols for secure computations. In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, pages 160–164. Citeseer, 1982.

[31] S. Zerr, E. Demidova, D. Olmedilla, W. Nejdl, M. Winslett, and S. Mitra. Zerber: r-confidential indexing for distributed documents. In *Proc. of EDBT*, pages 287–298, 2008.