Journal of Ubiquitous Systems & Pervasive Networks
Volume 15, No. 2 (2021) pp. 43-53

An Effective and Efficient Framework for Fast Privacy-Preserving Keyword Search on Encrypted Outsourced Cloud Data

Alfredo Cuzzocrea[a] *, Carson Leung[b], S. Sourav[b], Bryan H. Wodi[b]

[a]*iDEA Lab, University of Calabria, Rende, Italy & LORIA, Nancy, France*
[b]*Department of Computer Science, University of Manitoba, Winnipeg, MB, Canada*

Abstract

Cloud providers offer storage as a service to the data owners to store emails and files on the cloud server. However, sensitive data should be encrypted before storing on the cloud server to avoid privacy concerns. With the encryption of documents, it is not feasible for data owners to retrieve documents based on keyword search as they can do with plain text documents. Hence, it is desirable to perform a multi-keyword search on encrypted data. To achieve this goal, we present a *fast privacy-preserving model for keyword search on encrypted outsourced data* in this paper. Specifically, the model first performs a keyword search on encrypted data and checks its support for dynamic operations. Based on keyword search results, it then sorts all the relevant data documents using the number of keywords matched for a given query. To evaluate its performance of our model, we applied the standard metrics like precision and recall. The results show the effectiveness of our privacy-preserving keyword search on encrypted outsourced data.

Keywords: *Big Data, Big Data Privacy, Big Data Security, Encryption, Privacy-Preserving Keyword Search, Privacy-Preserving Information Retrieval*

1. Introduction

Nowadays, *big data management and analytics* are gaining momentum within the research community (e.g., [1-3]). Basically, the main issue with big data management concerns with effectively and efficiently managing massive big data repositories for a wide variety of typical data management tasks, such as representation, querying, indexing, partitioning, and so forth. All these data management tasks are recognized within the broad context of a hypothetical *big data management server*, which would be the forerunner of classical and well-consolidated DBMS servers, whose technology is mature and solid at now. On the other hand, big data analytics concerns with extracting useful, actionable knowledge from big data repositories for decision making purposes, by extending classical approaches inherited from decades of data mining and machine learning research (e.g., [4]), in a wide range of application scenarios ranging from social networks to bio-informatics, from sensors networks to web recommendation tools, from *e*-science systems to *e*-government systems, and so forth.

Distributed environments are the natural humus for big data management and analytics tasks. Among others, Cloud systems play the major role, even stirred-up by recent technological advancements that have really enhanced the ICT industry at now. More and more today, real-life Cloud-based applications, such as smart cities, intelligent transportation systems, marketplace tools and so forth, are indeed posing new challenges to big data research, thus contributing to improve the scientific area.

Big data management and big data analytics in distributed environments well converge within a common, unifying context whose main issues and challenges ask for common solutions to globally address the annoying problem of managing and supporting knowledge discovery from massive amounts of data.

In this so-delineated context, the top-class topic *privacy-preserving big data management and analytics in distributed environments* is emerged in the literature. It should be noted that this topic not only introduces relevant challenges at the theoretical level, but also is reminiscent of significant pragmatic advancements in real-life Cloud-based applications and systems. To become convinced of this, consider, for instance, the clear case represented by actual *bio-informatics systems* (e.g., [5]), where such issues assume a prominent role. The latter because of, generally, such systems store large amounts of *personal data*.

This critical topic is now heavily influencing the research community, and will play more and more a first-class role in actual and future research experiences (e.g., [88, 89, 90]).

* Corresponding author. Tel.: +390984492546
Fax: +390984492598; E-mail: alfredo.cuzzocrea@unical.it
This research has been made in the context of the Excellence Chair in Computer Engineering – Big Data Management and Analytics at LORIA, Nancy, France

DOI: 10.5383/JUSPN.15.02.006

1.1. Data Science Techniques for Supporting Big Data Management and Analytics

Thanks to technological advancements, data are everywhere at now. To elaborate, huge volumes of a wide variety of value data—which may be of different levels of veracity (e.g., precise or uncertain data)—can be easily generated or collected at a high velocity from wide ranges of rich sources of data in various real-life big data applications and services. Embedded in these big data are useful information and valuable knowledge. Hence, *data science* [27, 28] is in demand. Typically, data science solutions focus on different aspects of the following characteristics (i.e., different *V*'s) of big data to help users to visualize and validate the extracted information and discovered knowledge:

- huge *Volume* of big data, which focuses on the quantity of data;
- wide *Variety* of big data, which focuses on differences in types, contents, or formats of data (e.g., images [29]);
- *Value* of big data, which focuses on the usefulness of data (e.g., knowledge that can be discovered from the big data);
- different levels of *Veracity* of big data, which focuses on the quality of data (e.g., precise data, uncertain and imprecise data [30, 31]);
- high *Velocity* of big data, which focuses on the speed at which data are collected or generated (e.g., data streams [30]);
- *Visualization* of big data, which focuses on how to represent the data, information and/or knowledge in a comprehensive manner [32];
- *Validity* of big data, which focuses on how to interpret the data, information and/or knowledge.

In general, data science solutions apply the following techniques and methods to big data for data analytics:

- databases;
- data mining;
- information retrieval (e.g., for keyword or patent search [33, 34]);
- machine learning (e.g., deep learning [35, 36]);
- mathematical modelling;
- statistical techniques;
- visualization.

1.2. Privacy-Preserving Tasks

With big data, it is desirable to retrieve information from the big data, to mine and analyze the big data to discover new knowledge, as well as to publish the big data and their discovered knowledge. However, gathering and distributing these data might be legally prohibited due to widely-held privacy concerns [37]. If anonymity could be guaranteed, ground-breaking advances could be achieved, which may bring various real-life business models in terms of data analysis, services, and mashups. This is an essential task that must definitely be settled since a huge amount of data related to personal information is inevitably incorporated into the big data trend.

1.2.1. Privacy-Preserving Data Publishing (PPDP)

In a PPDP model [38, 39], data are published without disclosing identity of the data subjects. It allows researchers to conduct de-identification, which removes the relationship between the data subjects and the identified data. In other words, PPDP focuses on what data can be published—and how they can be published—without disclosing identity of the data subjects. Usually, *K*-anonymity [40] (via techniques like suppression, generalization, clustering, obfuscation, etc.) is relevant to de-identification so that it tries to protect privacy by employing the number of records having the same sequences in the trajectory database.

As an example, Eom *et al.* [38] recently presented an effective privacy-preserving data publishing model—which balances data utility and privacy preservation—based on vectorization (specifically, surrogate vectors). The model protects the private location information of individuals, and is applicable on grid environments.

1.2.2. Privacy-Preserving Data Mining (PPDM)

In a PPDM model, data that can be used for machine learning or statistical processing are not published, but are released as a form of statistical summary or as results based on aggregation, calculation, etc. In other words, PPDM focuses on what data can be released as a form of data mining results (e.g., summary of discovered knowledge)—and how they can be released—without disclosing identity of the data subjects [41]. Usually, differential privacy [42] prevents individual information disclosure based on a mathematical definition by adding noises.

As an example, in [43] we a privacy-preserving item-centric data mining algorithm that helps users to discover frequent patterns for big data. The algorithm allows users to express their preferred level of privacy, including:

- *k*-anonymity [44];
- *l*-diversity [45] (e.g., distinct *l*-diversity, entropy *l*-diversity, recursive *(c-l)*-diversity);
- *t*-closeness [46].

By doing so, the algorithm release summaries of data mining results that satisfy the user-preferred level of privacy.

1.2.3. Privacy-Preserving Data/Information Retrieval Based on Keyword Search

In the present world, data owners generate a lot of data using various applications and involve a lot of sensitive information about an individual and organization. In the last decade, the data owner was able to store a limited amount of data on the local machine using the hard disk, DVDs, floppy disk etc. To manage this data for utility purposes, the management and integrity of data are required which ensures effectively storage and retrieval of the data [47] using local machines but involves a large amount of cost in terms of time and hardware.

With the advent of cloud computing, the data owner and organization are motivated to store their data on cloud infrastructure without purchasing computational resources to manage the data by their own and access ubiquitous services with less operational overhead [49].

Although cloud services have various advantages, sensitive information of an organization, individual and health records possess privacy concern in cloud infrastructure that resists data owner to use cloud services. However, the *Cloud Service Providers* (CSPs) provide privacy-preserving solutions [48]. But still, it is prone to attack from malicious inside attackers that utilize the information of an individual without any authorization and abuse it. Moreover, outside attacks also compromised data confidentiality. A simple approach to solving this problem of confidentiality is to encrypt the data of the data owner before outsourcing it on the cloud server [50]. However, computation problems like search, update and delete are very difficult when the data stored on a cloud is encrypted. The only trivial solution is downloading all the encrypted data from the cloud server and decrypt it locally to perform various computations, but it is very impractical because it requires a large amount of network bandwidth and hardware cost.

Hence, it is of paramount importance to perform some searching operation on encrypted data based on multiple keywords without losing data confidentiality. Specifically, it is desirable to build search mechanism for data retrieval that offers result relevance ranking for relevant documents in a cloud server based on a given input query and provides data consumers with most relevant data documents in sorted order from all searched documents [51].

1.3. Privacy-Preserving Big Data Management and Analytics in Distributed Environments

In this investigated context, the issue of supporting *privacy-preserving* big data management and analytics (e.g., [6-8]) plays a first-class role, especially with respect to the wide class of emerging big data application scenarios, which range from social networks to bio-informatics, from sensors networks to web recommendation tools, from *e*-science systems to *e*-government systems, and so forth. In all these applicative settings, protecting the privacy of *sensitive information*, for instance personal data (e.g., [9-11]) or aggregate data (e.g., [12-14]), can be clearly intended as an *enabling technology*. Other emerging topics include the astonishing raise of *blockchain* technology (e.g., [15]).

On the other side, privacy-preserving big data management and analytics is strictly related to the research area recognized as *big data security* (e.g., [16]), which investigates how to securely access and handle big data repositories. The issue of combining *privacy and security of big data* (e.g., [17]), still in distributed environments, is, not by chance, one of the so-called ”hot-topics" directions for big data research of the future.

Following the great deal of interest for privacy-preserving big data management and analytics in distributed environments that has emerged during the last years (e.g., [10, 18, 22]), the research community already exposes quite a large literature on the topic. This demonstrates the maturity of the topic as well. Where future efforts will be oriented to? This paper aims at answering to this challenging question. From a side, it is un-doubtful that *theoretical tools* for supporting privacy-preserving big data management and analytics in distributed environments represent a very interesting research area to be explored. In this context, extending well-consolidated theoretical models for *privacy-preserving OLAP* (e.g., [18-20]) to emerging tools such as *differential privacy* (e.g., [21]) is a promising research direction. This paradigm is further sensible to be extended to more general *privacy-preserving big data publishing problems* (e.g., [22]) whose integration with innovative advanced machine learning tools, such as *tensor-based big data analytics* (e.g., [23]), constitutes a vibrant area of research with outstanding outcomes in both theoretical contributions and practical achievements. On the other hand, as regards big data analytics properly, another interesting line of research for the investigated area is represented by the issue of supporting *long-running big data analytics query processing in distributed environments* (e.g., [24]), for instance *Cloud stores* (e.g., [25]), in a privacy-preserving manner. Here, the main problem consists in how to *combine* the privacy preservation of *singleton* query (e.g., OLAP query – [26]) that composes the distributed big data analytics task with the privacy preservation of the *whole* distributed big data analytics task composed by (singleton) queries.

In the following, we report some possible guidelines for next-generation research in the context of privacy-preserving big data management and analytics in distributed environments:

- analysis of state-of-the-art proposals in the context of privacy-preserving big data management and analytics in distributed environments;
- definition of target privacy-preserving big data management and analytics scenarios in distributed environments to be used as case studies (e.g., Internet of Things, social networks, Cloud stores, etc.);
- definition of target privacy preserving big data management and analytics tools/processes in distributed environments to be addressed (e.g., OLAP, data publishing, tensor-based analytics, long-running big data analytics, etc.);
- definition of innovative privacy-preserving big data management and analytics tools in distributed environments, for instance based on differential privacy theory;
- design, implementation and testing of privacy-preserving big data management and analytics algorithms in distributed environments;
- definition and implementation of reference case studies for deeply assessing privacy-preserving big data management and analytics in distributed environments.

1.4. Contributions and Organization

In this paper, we design and implement a secure search scheme over the encrypted cloud data, which supports keyword ranked search on the data collection in the cloud. This is a fusion of theoretical design and practical implementation of a fast secure search scheme. With it, data owners will be able to insert and delete their encrypted data on the cloud as well as obtain high search efficiency using keywords. Our *key contributions* of this paper include the following:

- we extend a secure searchable encryption scheme that enables data owners to update encrypted data on the cloud;
- we implement a parallel search process on encrypted data to further reduce the time cost.

The remainder of this paper is organized as follows. The next described related works. Then, we formulate our problem, explain our theoretical design in Section 3, and present the methodology behind our practical implementation in Section 4. Evaluation results are shown in Section 5. Conclusions are drawn in Section 6. The conference version of this paper appears in [87].

2. Related Work

The security of data is a paramount concern for the cloud service provider. CSPs are responsible for building a secure business model that allows processing, formatting, and transmitting individuals’ data to the remote location while protecting it from external and internal threats [52]. In the public cloud, different users have unauthorized access to the data. Hence, a policy that helps protect against unauthorized access by different users is required. CSPs offer various isolation techniques to solve the problem of unauthorized access and provide encryption, access control, virtualization solutions to the data owners for data dissemination [53].

Secure keyword matching problem is very important in the cloud environment and gains a lot of attention from the community involved in secure communication. As a result, the following methods have been proposed to solve this problem:

- *single keyword searchable encryption* (see Section 2.1);
- *Boolean keyword searchable encryption* (see Section 2.2);

- *multi-keyword ranked searchable encryption* (see Section 2.3).

2.1. Single Keyword Searchable Encryption

There are various single keyword techniques available to search on encrypted data [54-56]. All these techniques stored encrypted data documents and its encrypted searchable index on the server using symmetric encryption. The data users can search a single keyword with the help of trapdoor that is built using the secret key offered by the data owner. For instance:

- Boneh *et al.* [57] created a public key technique where all users with the public key can write data on the server and users with the private key can perform search operation;
- Katz *et al.* [58] presented an algorithm for secure DNA pattern matching;
- Mohassel *et al.* [59] proposed an algorithm for Discrete Finite Automaton (DFA) evaluation.

These algorithms based on public key encryption have a very high computational expense and involve extra cost while search operation.

Li *et al.* [60] proposed a technique to search keyword from encrypted data. It follows similarity based keyword matching and maintains an error correction technique that is used to handle errors or "typos" by users. These techniques are based on predefined hamming distance and output keywords that are maintained in a query string following distance metric. But, it is not possible that hamming distance is not reliable for measuring the similarity of string pattern query because matching keywords might have a variable length.

2.2. Boolean Keyword Searchable Encryption

There are some techniques that follow conjunctive and disjunctive search over the encrypted data [61, 62]:

- The conjunctive search involves returning either all or nothing. This means that, if *all* search keywords mentioned in a query are present in some encrypted documents, then it will return all those documents having all same exact keywords. Otherwise (i.e., some search keywords mentioned in a query are absent from encrypted documents), it will not return the documents.
- The disjunctive approach involves returning all those documents, which can either have a subset of a specific keyword in a search query.

They have high computational cost because of bilinear mapping. Moreover, these techniques do not support multiple keyword searches on encrypted data stored on the cloud server.

Table 1. Different Techniques Used in Searchable Encryption

Method	Symmetric Key	Public Key	Index	Rank	Updates
Single-Keyword Search	[54-56]	[57-59]	[54-56]	X	X
Boolean Search	X	[61-62]	X	X	X
Multi-Keyword Search	[64, 69]	X	[64, 69]	[64, 69]	[69]

2.3. Multi-Keyword Ranked Searchable Encryption

Ranked search helps data users to utilize the most relevant documents based on the query string. These algorithms return top-k most relevant documents and decrease network traffic while serving the request based on the input query string. For instance:

- Swaminathan *et al.* [63] created ranked based searchable technique but it was designed for a query involving only a single keyword search.
- Cao *et al.* [64] developed a multi-keyword ranked search on encrypted data. They represented documents and queries as a vector of dictionary size and perform coordinate matching. It returns relevant documents in sorted order based on the number of matched keywords for a given input query string.
- Sun *et al.* [65] also created a secure multi-keyword search scheme. This scheme was based on similarity-based ranking and used a searchable index tree based on the vector space model. For ranking the results, they used cosine along with *Term Frequency* (TF) × *Inverse Document Frequency* (IDF).

However, these techniques do not work when the data owner wants to perform an update or delete operation on a given dataset on the cloud server. Hence, there is a need for the searchable encryption technique that supports dynamic operations. Such kind of operations are those referred to data and indexes that dynamically change over time.

Different approaches discussed in related work for searchable encryption having different characteristics are described in Table 1.

2.4. Other Approaches

Hu *et al.* [66] proposed an *index blind storage* (*IBS*) *mechanism*, which enables one perform range queries on encrypted cloud data while concealing the query pattern. This enhances the security of the search process. The general idea is storing a key hash of the plain text in a separate column on the table and generating an index on that column. Moreover, Hu *et al.* [66] recommended to have *separate encryption key and index key*, and store them on the web server of the application. This way, the database server would have no way of obtaining the keys. Hence, if an adversary were to break into the database, then they would not be able to learn anything meaningful from the data except its access and search patterns [67]. This makes our approach safe from outsider attack and insider attack when it relates to the cloud service provider. The only way a third party can breach this model is if and only if they have access to the symmetric encryption key used in the encryption of the owner's files, which will only be possible if they physically break into the data owner's or data user's computer.

3. Design of Our Privacy-Preserving Keyword Search Model

3.1. System Model

The architecture of ranked search over encrypted outsourced data in cloud storage is illustrated in Fig. 1. Within the architecture, the three key entities involving in this problem include the following:

- the data owner;
- the cloud service provider;
- the data user.

Let us elaborate on each of these key entities.

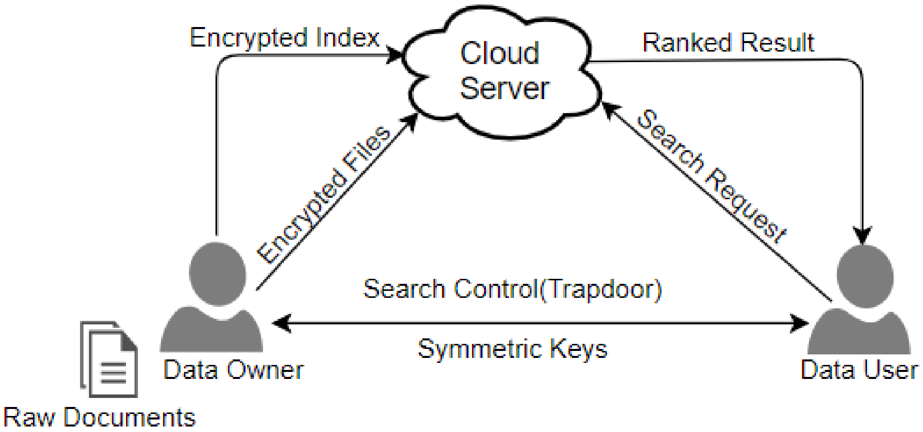

Fig. 1. The architecture of ranked search over encrypted outsourced data in cloud storage

3.1.1. Data Owner

In general, a **data owner** has a collection of various raw documents $\mathcal{F}$ that he wishes to store at remote cloud location in the encrypted format. In order to effectively utilize these encrypted files for the search operation, the data owner makes the following tasks:

- first, building an encrypted tree index $\mathcal{I}$ for all the keywords gather from various documents $\mathcal{F}$;
- then, generating cipher-text collection $\mathcal{C}$ for all the documents $\mathcal{F}$;
- also, outsourcing both cipher-text collection $\mathcal{C}$ and encrypted index $\mathcal{I}$ to the cloud storage server.

However, the data owner has provision to update various documents in the cloud server by encrypting them locally and send it to the server later on.

3.1.2. Cloud Service Provider

Generally, a **cloud service provider** stores both of the following in the remote location:

- cipher-text collection $\mathcal{C}$;
- encrypted tree index $\mathcal{I}$.

Then, the cloud service provider will take the following actions:

- receives the search request from the data user in the form of trapdoor *TD*;
- executes the search operation on encrypted tree index based on *TD*;
- returns relevant encrypted documents with rank order k to the data user;
- performs update operation for various new documents provided by the data owner and the encrypted tree index $\mathcal{I}$.

3.1.3. Data Users

In general, **data users** are authorized by the data owner to obtain a shared secret key for utilizing encrypted data. Data users can use the search control mechanism to:

- perform search query using t keywords by using trapdoor *TD*;
- retrieve k encrypted documents from the cloud storage server based on the query keyword.

As a preview, the cloud server used in this project is considered as "honest-but-curious". This means that the cloud server applies the following actions:

- performs all the operations in an honest manner;
- nevertheless, it is curious in the sense that it tries to:
  - infer the knowledge from query data;
  - analyze the inferred knowledge to gain additional information;
  - perform various operations.

3.2. Threat Model

In this paper, we consider two threat models [64], which are based on the information acquired by the cloud server:

- *Known cipher-text model*, in which the cloud server has knowledge of both cipher-text collection $\mathcal{C}$ and the encrypted tree index $\mathcal{I}$ that are outsourced to the cloud by the data owner. In this model, the cloud server is capable of performing on the cipher-text only attack.
- *Known background model*, in which the cloud server is stronger when compared to known cipher-text model because the cloud server has more access to knowledge regarding cipher-text collection. The information involves term frequency (TF) and statistics about the number of documents for each keyword in the whole documents. In this kind of

attack, the cloud server tries to deduce the query keyword based on keyword frequency.

3.3. Design Goals

The system model based on above design properties enable secure, accurate and dynamic keyword ranked search over encrypted data stored in cloud server and the system design tries to achieve following three goals. These are described in the following Sections.

3.3.1. Dynamic Multi-Keyword Ranked Search

Our designed system aims to perform:

- the multi-keyword query on encrypted data stored in the cloud server;
- the dynamic update of documents on the cloud server.

3.3.2. Efficiency

Our proposed scheme ensures efficiency query mechanism by following index based tree that offers sub-linear efficiency while searching.

3.3.3. Privacy Preserving

Our proposed scheme is designed in a way that cloud server does not learn additional information about the stored documents, search-able tree index and query but only learn about search result that is returned by cloud server after performing the query. The proposed scheme is designed to meet the following privacy preserving requirements:

- *Trapdoor Unlinkability*: The function of generating trapdoor should not be deterministic because the adversary or cloud server will be able to deduce whether the two trapdoors are generated for the same query keyword or not. Using a deterministic approach, the cloud server will be able to generate the frequencies of different keywords used in the different search query that will breach the privacy requirement of the underlying keywords in the search query. The trapdoor should be generated using a non-deterministic approach so that it will ensure privacy requirement for the same keyword that is utilized for the same search query.
- *Keyword Privacy*: The main concern regarding privacy is to hide the underlying keyword used for the search. The trapdoor function produces an encrypted search query for a given keyword and protects it from the adversary. But adversary can do some statistical inference to predict the keyword of the search query. And, if the adversary knows background information about known cipher-text model, it will be utilized to find out the keyword of the search query.
- *Index Confidentiality and Query Confidentiality*: The raw documents, keywords in the index and query are encrypted so that adversary does not know anything about data.

4. Implementation of Our Privacy-Preserving Keyword Search Model

Our approach is, to begin with, implementing literal search of encrypted data for single keywords and then expanding its utility to perform partial match searching. When we achieve this, we further extend it to accommodate multiple keywords. We aim to achieve this by employing the blind indexing strategy (see Section 2.4).

4.1. Blind Indexing

Recall from Section 2.4 that an *index blind storage* (*IBS*) *mechanism* enables one perform range queries on encrypted cloud data while concealing the query pattern. This IBS mechanism enhances the security of the search process. The general idea is storing a key hash of the plain text in a separate column on the table and generating an index on that column. For the files uploaded on the cloud in this project, the construction of our table is shown in Fig. 2. Here, we create a *fileUpload* table and also create its associated *keyWordIndex* index. More specifically, we store a blind index of the plain keywords in *keyWordBlindIndex*. Then, the keyword is encrypted and stored on the table.

```
CREATE TABLE IF NOT EXISTS fileupload
(
 id INT NOT NULL AUTO_INCREMENT,
 uname TEXT NOT NULL,
 filename VARCHAR(255) NOT NULL UNIQUE,
 filesize VARCHAR(255) NOT NULL,
 filePath VARCHAR(255) NOT NULL,
 keyWord TEXT NOT NULL,
 indexVal VARCHAR(255) NOT NULL,
 keyWordBlindIndex VARCHAR(255),
 PRIMARY KEY (id)
);
CREATE INDEX keyWordIndex ON
fileupload(keyWord);
```

Fig. 2. The *fileUpload* table and the associated *keyWordIndex* index

Afterwards, we adapt Java's *PBKDF2WithHmacSHA1* algorithm for *password-based key derivation function* (*PBKDF*) *2 with hash-based message authentication code (HMAC) using the secure hash algorithm 1* (*SHA1*), in which a combination of hashing and a unique salt is applied in order to derive the key from the keyword. The resulting algorithm, as outlined in algorithm *getListDocumentsWithKeyword* (see Figure 3), returns a list of documents that contain that keyword. With algorithm *getListDocumentsWithKeyword*, we are able to successfully query the encrypted database for exact matches of the keywords.

```
Algorithm getListDocumentsWithKeyword
Input: Collection of documents, keyword.
Output: List of documents containing that keyword.
Begin
   1. blindIndex ← getKeyWordBlindIndex(keyword,
blindIndexKkey);
   2. sqlStmt ← PreparedStatement(  SELECT *
                  FROM fileupload
                  WHERE keyWordBlindIndex =
                            blindIndex);
   3. result ← sqlStmt.execute().getResultSet();
   4. return result;
End;
```

Fig. 3. Algorithm *getListDocumentsWithKeyword*

To further enhance the searching process for privacy-preserving information retrieval by accommodating partial string matches, we create a new table called *fileUploadFilter* as outlined in Fig. 4. We also created its associated index *keyWordFilterIndex* on (*stringMatch*, *trapdoor*) to speed up the searching process for privacy-preserving information retrieval.

```
CREATE TABLE IF NOT EXISTS fileupload_filter
(
 filterID INT NOT NULL AUTO_INCREMENT,
 fileUploadID INT,
 stringMatch TEXT,
 trapdoor TEXT,
 PRIMARY KEY (filterID),
 FOREIGN KEY (fileUploadID) REFERENCES
fileupload(ID)
);
CREATE INDEX keyWordFilterIndex ON
fileupload_filter(stringMatch,trapdoor);
```

Fig. 4. The *fileUploadFilter* table and the associated *keyWordFilterIndex* index

Next, we store distinct blind indexes per column for every type of query we needed. For single keywords, we store a blind index of the first five characters for every keyword the user enters so we can perform a search of the following form shown in Fig. 5.

```
SELECT *
FROM fileupload
WHERE keyWord LIKE '%cloud%'
```

Fig. 5. Search query example

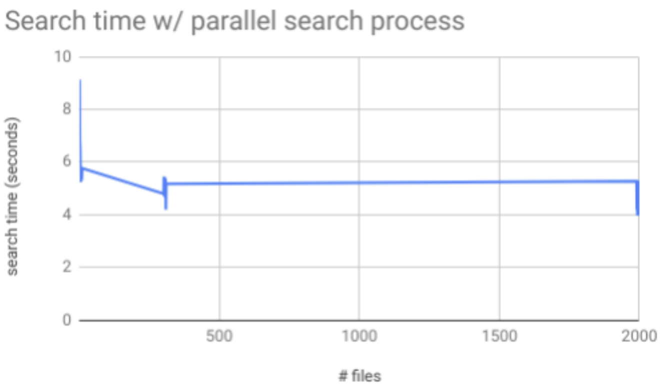

Fig. 6. Search time *with* parallel search process

Fig. 7. Search time *without* parallel search process

4.2. Analysis

Recall from Section 2.4, it was recommended to have separate encryption key and index key, and store them on the web server of the application. This way, the database server would have no way of obtaining the keys. Hence, if an adversary were to break into the database, then they would not be able to learn anything meaningful from the data except its access and search patterns. This makes our approach safe from outsider attack and insider attack when it relates to the cloud service provider (CSP). The only way a third party can breach this model is if and only if they have access to the symmetric encryption key used in the encryption of the files owners, which will only be possible if they physically break into the data owner's or data user's computer.

Consequently, in our table, we added duplicate entries for the possible partial search strings. The reason we did this was to aid indexing, which allows for fast SELECT queries. To reduce memory consumption, we chose to create one index for partial SELECT queries matching the first five letters.

We also boosted the performance further by truncating the blind indexes to 16 bits and used them as *Bloom Filter*. The benefits we noticed was the improved speed in getting the results from queries.

5. Evaluation: Experiments and Discussion

For evaluation, we compared with closely related work. For instance, Cao *et al.* [64] created a non-dynamic multi-keyword search scheme where data owners could not update their data once uploaded on the cloud; they could only perform multi-keyword searches. Also, Sun *et al.* [65] improved on that scheme by providing the update functionality. The problem with their approach was that they were performing a lot of computation on the cloud to re-generate the index once a change is made. This made the performance of the application to degrade with more dynamic operations.

Our desired approach was to store a separate, distinct blind index per column for every different kind of query we need (each with its own key), instead of regenerating the index.

For our experiment, we computed search times for multi-word queries on a collection of 2,000 unencrypted documents. While this document quota is reasonable for a complete preliminary evaluation, further experimental exploration on the performance of our technique is left as future work We fed the contents of the documents to a *Text Analyzer* [70]. The keywords were made using a random word generator from a collection of five words. We measured the search times for queries *using our parallel scheme* and compared them to the run time for the same queries *without the scheme*. We also measured the percentage of RAM and thread loads for both scenarios. The results can be seen in Figs. 6–9.

The experiments were run on a *32GB Intel Core i7-7700HQ CPU @ 2.80GHz* processor with 8 threads and a *GeForce GTX 1060* graphics card. We used *DropBox* as a Cloud Service Provider and performed upload and retrieval of files using API calls.

Observed from Figs. 6–7, when the queries are run in parallel, search time stays relatively at 5 seconds on average. In contrast, they tend to grow proportional to the number of files when run sequentially showing that our improvement has potential of improving the multi-keyword search on encrypted data. Figs. 8–9 show the trade-off of this design.

6. Conclusions and Future Work

We proposed a fast privacy-preserving keyword search model on encrypted outsourced data. Specifically, our model performs dynamic search on encrypted data located in the cloud. In our model, we implemented a rank documents feature that supports querying the frequency of encrypted keywords on the cloud. We also maintained the privacy and data integrity by employing a symmetric encryption/decryption scheme for transferring and querying encrypted content on the cloud. As an added contribution, we implemented a parallel search scheme to increase search query speed. Based on the symmetric key encryption, we guarantee that it is invulnerable to attack unless one physically gains access to the key which is not possible in the ideal world.

As ongoing work, we are exploring further improvements to our privacy-preserving keyword search. For instance, we are exploiting user-specified constraints [71, 72] and incorporating them into the search [73, 74]. We are investigating the benefits of having visualization [75] to our approach. For future work, we plan to incorporate additional knowledge—such as information from online analytical processing (OLAP) data [76-79], graphs [80-84], web [85, 86]—into our approach for performing more effective searches.

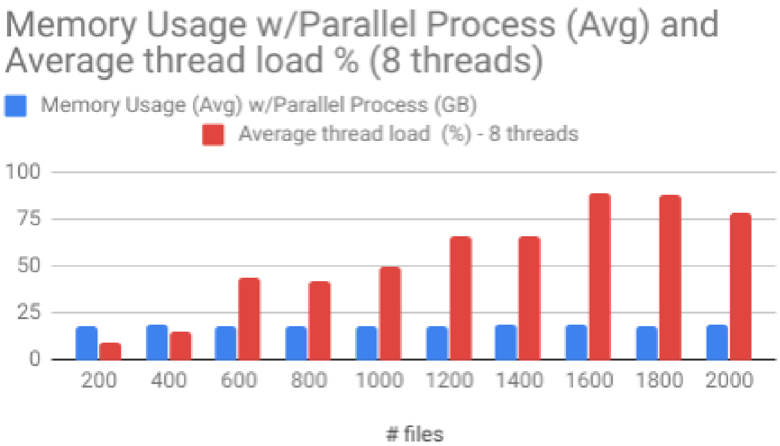

Fig. 8. Memory utilization and thread load *without* parallel search process

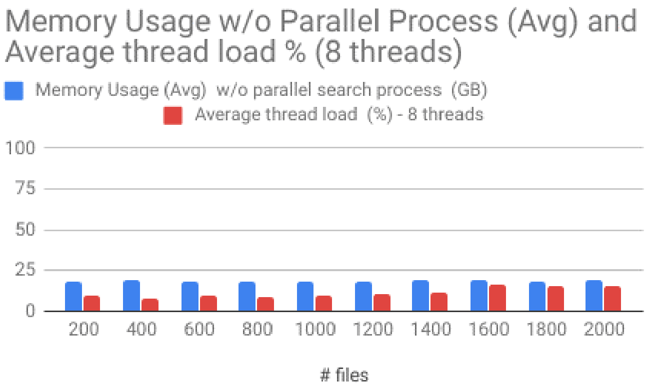

Fig. 9. Memory utilization and thread load *witouth* parallel search process

Acknowledgments

This research has been partially supported by the following projects:

- Natural Sciences and Engineering Research Council of Canada (NSERC) – Canada;
- University of Manitoba – Canada;
- PIA project "Lorraine Université de Excellence" reference ANR-15-IDEX-04-LUE – France.

References

[1] D. Agrawal, S. Das, A. El Abbadi, "Big Data and Cloud Computing: Current State and Future Opportunities," in Proc. EDBT 2011, pp. 530-533.

[2] A. Labrinidis, H.V. Jagadish, "Challenges and Opportunities with Big Data," Proceedings of the VLDB Endowment 5(12), pp. 2032-2033, 2012.

[3] P. Zikopoulos, C. Eaton, "Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data," McGraw-Hill Osborne Media, 2011.

[4] A. Cuzzocrea, "Combining Multidimensional User Models and Knowledge Representation and Management Techniques for Making Web Services Knowledge-Aware," Web Intelligence and Agent Systems 4(3), pp. 289-312, 2006.

[5] N. Mehta, A. Pandit, S. Shukla, "Transforming Healthcare with Big Data Analytics and Artificial Intelligence: A Systematic Mapping Study," Journal of Biomedical Informatics 100, 2019.

[6] R. Lu, H. Zhu, X. Liu, J.K. Liu, J. Shao, "Toward Efficient and Privacy-Preserving Computing in Big Data Era," IEEE Network 28(4), pp. 46-50, 2014.

[7] H.-Y. Tran, J. Hu, "Privacy-Preserving Big Data Analytics A Comprehensive Survey," Journal of Parallel and Distributed Computing 134, pp. 207-218, 2019.

[8] A. Cuzzocrea, E. Damiani, "Making the Pedigree to Your Big Data Repository: Innovative Methods, Solutions, and Algorithms for Supporting Big Data Privacy in Distributed Settings via Data-Driven Paradigms," in Prof. IEEE COMPSAC 2019, pp. 508-516, 2019.

[9] P. Liang, L. Zhang, L. Kang, J. Ren, "Privacy-Preserving Decentralized ABE for Secure Sharing of Personal Health Records in Cloud Storage," Journal of Information Security and Applications 47, pp. 258-266, 2019.

[10] M.H. Au, K. Liang, J.K. Liu, R. Lu, J. Ning, "Privacy-Preserving Personal Data Operation on Mobile Cloud - Chances and Challenges over Advanced Persistent Threat," Future Generation Computer Systems 79, pp. 337-349, 2018.

[11] E.G. Komishani, M. Abadi, F. Deldar, "PPTD: Preserving Personalized Privacy in Trajectory Data Publishing by Sensitive Attribute Generalization and Trajectory Local Suppression," Knowledge Based Systems 94, pp. 43-59, 2016.

[12] A. Cuzzocrea, "Privacy-Preserving Big Data Management: The Case of OLAP," in â€œBig Data - Algorithms, Analytics, and Applications," Chapman and Hall/CRC, pp. 301-326, 2015.

[13] A. Cuzzocrea, D. Saccà , "A Constraint-Based Framework for Computing Privacy Preserving OLAP Aggregations on Data Cubes," in Proc. ADBIS 2011, pp. 95-106, 2011.

[14] A. Cuzzocrea, V. Russo, D. Saccà , "A Robust Sampling-Based Framework for Privacy Preserving OLAP," in Proc. DaWaK 2008, pp. 97-114, 2008.

[15] Z. Zheng, S. Xie, H. Dai, X. Chen, H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," in Proc. IEEE BigData Congress 2017, pp. 557-564.

[16] C. Tankard, "Big Data Security," Network Security 2012(7), pp. 5-8, 2012.

[17] A. Cuzzocrea, "Privacy and Security of Big Data: Current Challenges and Future Research Perspectives," in Proc. ACM CIKM 2014, pp. 45-47, 2014.

[18] A. Cuzzocrea, E. Bertino, "Privacy Preserving OLAP over Distributed XML Data: A Theoretically-Sound Secure-Multiparty-Computation Approach," Journal of Computer and System Sciences 77(6), pp. 965-987, 2011.

[19] A. Cuzzocrea, E. Bertino, D. Saccà , "Towards a Theory for Privacy Preserving Distributed OLAP," in ACM EDBT/ICDT 2012, pp. 221-226, 2012.

[20] A. Cuzzocrea, V. Russo, "Privacy Preserving OLAP and OLAP Security," in Encyclopedia of Data Warehousing and Mining 2009, pp. 1575-1581, 2009.

[21] C. Dwork, "Differential Privacy: A Survey of Results," in TAMC 2008, pp. 1-19, 2008.

[22] H. Zakerzadeh, C.C. Aggarwal, K. Barker, "Privacy-Preserving Big Data Publishing," in ACM SSDBM 2015, pp. 26:1-26:11, 2015.

[23] Q. Song, H. Ge, J. Caverlee, X. Hu, "Tensor Completion Algorithms in Big Data Analytics," ACM Transactions on Knowledge Discovery from Data 13(1), pp. 6:1-6:48, 2019.

[24] M. Qaosar, K.M.R. Alam, C. Li, Y. Morimoto, "Privacy-Preserving Top-K Dominating Queries in Distributed Multi-Party Databases," in IEEE BigData 2019, pp. 5794-5803, 2019.

[25] K. Grolinger, W.A. Higashino, A. Tiwari, M.A.M. Capretz, "Data Management in Cloud Environments: NoSQL and NewSQL Data Stores," Journal of Cloud Computing, art. 22, 2013.

[26] T. Wang, B. Ding, J. Zhou, C. Hong, Z. Huang, N. Li, S. Jha, "Answering Multi-Dimensional Analytical Queries under Local Differential Privacy," in ACM SIGMOD 2019, pp. 159-176, 2019.

[27] K. E. Dierckens, A. B. Harrison, C. K. Leung, and A. V. Pind, "A data science and engineering solution for fast k-means clustering of big data," in Proc. IEEE TrustCom-BigDataSE-ICESS 2017, pp. 925-932.

[28] C. K. Leung, F. Jiang, H. Zhang, and A. G. M. Pazdor, "A data science model for big data analytics of frequent patterns," in Proc. IEEE DASC-PICom-DataCom-CyberSciTech 2016, pp. 866-873.

[29] P. Buayai, K. Piewthongngam, C. K. Leung, and K. Runapongsa Saikaew, "Semi-automatic pig weight estimation using digital image analysis," Applied Engineering in Agriculture, 35(4), 521-534, 2019.

[30] J. A. Ovi, C. F. Ahmed, C. K. Leung, and A. G. M. Pazdor, "Mining weighted frequent patterns from uncertain data stream," in Proc. IMCOM 2019, pp. 917-936.

[31] M. M. Rahman, C. F. Ahmed, and C. K. Leung, "Mining weighted frequent sequences in uncertain databases," Information Sciences, vol. 479, 76-100, 2019.

[32] C. K. Leung and Y. Zhang, "An HSV-based visual analytic system for data science on music and beyond," International Journal of Art, Culture and Design Technologies (IJACDT), 8(1), 68-83, 2019.

[33] A. Cuzzocrea, W. Lee, and C. K. Leung, "High-recall information retrieval from linked big data," in Prc. IEEE COMPSAC 2015, vol. 2, pp. 712-717.

[34] C. K. Leung, W. Lee, and J. J. Song, "Information technology-based patent retrieval model," in Springer Handbook of Science and Technology Indicators, pp. 859-874, 2019.

[35] J. de Guia, M. Devaraj, and C. K. Leung, "DeepGx: deep learning using gene expression for cancer classification," in Proc. IEEE/ACM ASONAM 2019, pp. 913-920.

[36] C. K. Leung, P. Braun, and A. Cuzzocrea, "AI-based sensor information fusion for supporting deep supervised learning," Sensors, 19(6), 1345:1-1345:12, 2019.

[37] T. Li and N. Li, "On the tradeoff between privacy and utility in data publishing," in Proc. ACM KDD 2009, pp. 517-526.

[38] C. S. Eom, C. C. Lee, W. Lee, and C. K. Leung, "Effective privacy preserving data publishing by vectorization," Information Sciences, 2019. DOI: 10.1016/j.ins.2019.09.035

[39] B. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: a survey of recent developments," ACM Computing Surveys (CSUR), 42(4), 14:1-14:53, 2010.

[40] L. Sweeney, "k-anonymity: a model for protecting privacy," International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 10(5), 557-570, 2002.

[41] W. Gan, J. C. Lin, H. Chao, S. Wang, and P. S. Yu, "Privacy preserving utility mining: a survey," in Proc. IEEE BigData 2018, pp. 2617-2626.

[42] C. Dwork, "Differential privacy: a survey of results," in Proc. TAMC 2008, pp. 1-19.

[43] C. K. Leung, C. S. H. Hoi, A. G. M. Pazdor, B. H. Wodi, and A. Cuzzocrea, "Privacy-preserving frequent pattern mining from big uncertain data," in Proc. IEEE BigData 2018, pp. 5101-5110.

[44] J. Domingo-Ferrer, "k-anonymity," in Encyclopedia of Database Systems, p. 1585, 2009.

[45] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "l-diversity: privacy beyond k-anonymity," ACM Transactions on Knowledge Discovery from Data (TKDD), 1(1), 3:1-3:52, 2007.

[46] C. Dwork, "Differential privacy," in Proc. ICALP 2006, Part II, pp. 1-12.

[47] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," Journal of Internet Services and Applications, 1(1), 7-18, 2010.

[48] Y. Shi, "Data security and privacy protection in public cloud," in Proc. IEEE BigData 2018, pp. 4812-4819.

[49] K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," IEEE Internet Computing, 16(1), 69-73, 2012.

[50] S. I. Kamara and K. Lauter, "Cryptographic cloud storage," in Proc. FC 2010, pp. 136-149.

[51] A. Singhal, "Modern information retrieval: A brief overview," IEEE Data Engineering Bulletin, 24(4), 35-43, 2001.

[52] A. Tripathi, and A. Mishra, "Cloud computing security considerations," in Proc. IEEE ICSPCC 2011, pp. 981-985.

[53] B. Gupta, D. P. Agrawal, and S. Yamaguchi, eds. Handbook of research on modern cryptographic solutions for computer and cybersecurity. IGI Global, 2016.

[54] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Proc. IEEE &SP 2000, pp. 44:1-44:12.

[55] E. Goh, "Secure indexes," IACR Cryptology ePrint Archive, report 2003/216.

[56] Y. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in Proc. ACNS 2005, pp. 442-455.

[57] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in Proc. EUROCRYPT 2004, pp. 506-522.

[58] J. Katz and L. Malka, "Secure text processing with applications to private DNA matching," in Proc. ACM CCS 2010, pp. 485-492.

[59] P. Mohassel, S. Niksefat, S. Sadeghian, and B. Sadeghiyan, "An efficient protocol for oblivious DFA evaluation and applications," in Proc. CT-RSA 2012, pp. 398-415.

[60] J. Li, Q. Wang. C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," Proc. IEEE InfoCom 2010, pp. 441-445.

[61] J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," Journal of Cryptology, 26(2), 191-224, 2013.

[62] 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. EUROCRYPT 2010, pp. 62-91.

[63] A. Swaminathan, Y. Mao, G. Su, H. Gou, A. L. Varna, S. He, M. Wu, and D. W. Oard, "Confidentiality-preserving rank-ordered search," in Proc. ACM StorageSS 2007, pp. 7-12.

[64] N. Cao, C. Wang, M. Li, K. Ren, W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted

cloud data," IEEE Transactions on Parallel and Distributed Systems (TPDS), 25(1), 222-233, 2014.

[65] W. Sun, B. Wang, N. Cao, M. Li, W. Lou, Y. T. Hou, and H. Li, “Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking," in Proc. ACM ASIA CCS 2013, pp. 71-82.

[66] Y. Hu, W. Li, M. Ma, N. Cao, Y. Liu, Z. Qin, and J. Wang, “Toward complex search for encrypted cloud data via blind index storage," in Proc. IEEE ISPA/IUCC 2017, pp. 1-8.

[67] J. Li, D. Lin, A. C. Squicciarini, J. Li, and C. Jia, “Towards privacy-preserving storage and retrieval in multiple clouds," IEEE Transactions on Cloud Computing (TCC), 5(3), 499-509, 2017.

[68] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai, “Cryptography from anonymity," in Proc. IEEE FOCS 2006, pp. 239-248.

[69] Z. Xia, X. Wang, X. Sun, and Q. Wang, “A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," IEEE Transactions on Parallel and Distributed Systems (TPDS), 27(2), 340-352, 2016.

[70] Online-Utility.org, https://www.online-utility.org/

[71] L. V. S. Lakshmanan, C. K. Leung, and R. T. Ng, “The segment support map: scalable mining of frequent itemsets," ACM SIGKDD Explorations 2(2), 21-27, 2000.

[72] C. K. Leung, “Frequent itemset mining with constraints," in Encyclopedia of Database Systems, Second Edition, pp. 1531-1536, 2018.

[73] C. K. Leung, S. K. Tanbeer, and J. J. Cameron, “Interactive discovery of influential friends from social networks," Social Network Analysis and Mining, 4(1), 154:1-154:13, 2014.

[74] S. K. Tanbeer, C. K. Leung, and J. J. Cameron, “Interactive mining of strong friends from social networks and its applications in e-commerce," Journal of Oanizational Computing and Electronic Commerce (JOCEC), 24(2-3), 157-173, 2014.

[75] C. K. Leung and C. L. Carmichael, “Exploring social networks: a frequent pattern visualization approach," in Proc. IEEE SocialCom 2010, pp. 419-424.

[76] L. Bellatreche, A. Cuzzocrea, and S. Benkrid, “F&A: a methodology for effectively and efficiently designing parallel relational data warehouses on het- erogenous database clusters," in DaWaK 2010, pp. 89-104, 2010.

[77] L. Bellatreche, A. Cuzzocrea, and S. Benkrid, “Effectively and efficiently designing and querying parallel relational data warehouses on heterogeneous database clusters: the F&A approach," Journal of Database Management, 23(4), 17-51, 2012.

[78] M. Ceci, A. Cuzzocrea, and D. Malerba, “Effectively and efficiently support- ing roll-up and drill-down OLAP operations over continuous dimensions via hierarchical clustering," Journal of Intelligent Information Systems, 44(3), 309-333, 2015.

[79] A. Cuzzocrea and C. K. Leung, “Efficiently compressing OLAP data cubes via R-tree based recursive partitions," in Proc. ISMIS 2012, pp. 455-465.

[80] N. Ashraf, R. R. Haque, M. A. Islam, C. F. Ahmed, C. K. Leung, J. J. Mai, and B. H. Wodi, “WeFreS: weighted frequent subgraph mining in a single large graph," in Proc. ICDM 2019, pp. 201-215.

[81] A. Cuzzocrea and I. Song, “Big graph analytics: the state of the art and future research agenda," in DOLAP 2014, pp. 99-101, 2014.

[82] M. A. Islam, C. F. Ahmed, C. K. Leung, and C. S. H. Hoi, “WFSM-MaxPWS: an efficient approach for mining weighted frequent subgraphs from edge-weighted graph databases," in Proc. PAKDD 2018, Part III, pp. 664-676.

[83] S. P. Singh, C. K. Leung, F. Jiang, and A. Cuzzocrea, “A theoretical approach to discover mutual friendships from social graph networks," in Proc. iiWAS 2019. DOI: 10.1145/3366030.3366077

[84] F. M. Upoma, S. A. Khan, C. F. Ahmed, T. Alam, S. A. Zahin, and C. K. Leung, “Discovering correlation in frequent subgraphs," in Proc. IMCOM 2019, pp. 1045-1062.

[85] M. Fisichella, A. Stewart, A. Cuzzocrea, and K. Denecke, “Detecting health events on the social web to enable epidemic intelligence," in Proc. SPIRE 2011, pp. 87-103.

[86] C. K. Leung, F. Jiang, and J. Souza, “Web page recommendation from sparse big web data," in Proc. IEEE/WIC/ACM WI 2018, pp. 592-597.

[87] B. H. Wodi, C. K. Leung, A. Cuzzocrea, S. Sourav, “Fast Privacy-Preserving Keyword Search on Encrypted Outsourced Data,” in Proc. IEEE BigData 2019, pp. 1-10.

[88] V. R. Q. Leithardt, D. Nunes, A. G. de Moraes Rossetto, C. Oberdan Rolim, C. F. R. Geyer, J. Sá Silva, “Privacy Management Solution in Ubiquitous Environments Using Percontrol,” Journal of Ubiquitous Systems and Pervasive Networks 5(2), 21-28, 2014.

[89] M. Talha, N. Elmarzouqi, A. A. El Kalam, “Quality and Security in Big Data: Challenges as opportunities to build a powerful wrap-up solution,” Journal of Ubiquitous Systems and Pervasive Networks 12(1), 9-15, 2020.

[90] Y. Issaoui, A. Khiat, A. Bahnasse, H. Ouajji, “Smart Logistics: Blockchain trends and applications,” Journal of Ubiquitous Systems and Pervasive Networks 12(2), 9-15, 2020.