An Effective and Efficient Multi-Keyword Ranked Searchable Security using FP-Growth Algorithm

K Madan Mohan, Research Scholar, Dept of Computer Science and Engineering, JNTUH, Kukatpally, Hyderabad,500085, TS.

Dr.B.V.Ram Naresh Yadav, Associate Professor, Department of Computer Science &Engineering, JNTUH College of Engineering Jagtial (JNTUHCEJ) ,Nachupally, Jagtial,505 501,TS.

Abstract:

Cloud computing could bring security problems. In order to ensure data security and user privacy, people would choose to store data in the cloud with ciphertext. How to search data efficiently and comprehensively without decryption has become the focus of this paper. Cloud computing provides individuals and enterprises massive computing power and scalable storage capacities to support a variety of big data applications in domains like health care and scientific research, therefore more and more data owners are involved to outsource their data on cloud servers for great convenience in data management and mining. So far, most of the works have been proposed under different threat models to achieve various search functions, such as single keyword search, similarity search, multi- keyword Boolean search, ranked search, multi-keyword ranked search, etc. Among them, multi- keyword ranked search achieves more attention for its practical applicability. propose a secure and ranked multi-keyword search protocol in a multi-owner cloud model over encrypted cloud data, which simultaneously supports dynamic update operations like deletion and insertion of documents.

In this paper, we propose an efficient privacy protection scheme. In this scheme, Elliptic Curve Cryptography (ECC) is adopted to encrypt the data. It can reduce the computing cost of encryption and decryption uploading the encrypted files and indexes to the cloud server. Then it can authorize users to generate trap door using hash conflict function, and send it to Cloud Service Provider (CSP) for searching for matched ciphertext. The CSP uses the FP-Growth algorithm to extend keywords and search index to match the ciphertext. In this paper, we will use the FP-Growth algorithm to extend the keywords' semantics, match the index list based on these keywords, and return the requested file-set which is more consistent with the user's search. Experiments show that compared with traditional methods, files can be encrypted, decrypted, and recovered more quickly when we use this method. It can also ensure the privacy of data and reduce the communication overhead.

Keywords: Big data, Cloud Computing, Elliptic Curve Cryptography (ECC), Cloud Service Provider (CSP), FP-Growth.

1. INTRODUCTION

A Computing application where in cloud clients can remotely keep their statistics into the cloud a good way to revel in the on-demand excessive satisfactory applications and offerings from shared pool of configurable computing assets is Cloud Computing. It is a form of computing that relies on sharing computing assets in place of having nearby servers or non-public devices to deal with packages. In simple terms, storing and gaining access to information and applications over the internet in place of a laptop's tough drive is the way in cloud computing. In present days, large information is common ordinary on-line. New and additional records are outsourced due to boom in garage plus requirements of customers daily, then essentially semi-relied on servers. Cloud customers can deliver their facts into the cloud [1] as Cloud computing is a web-primarily based model. The facts proprietors live unbound after the potential of storage by loading information into the cloud. It is a vital challenge to guard sensitive facts integrity. The records owner has to be outsourced within the encoded device to the general public cloud and the information operation is based on plaintext keyword seek to safeguard facts private ness within the cloud. We use the green degree of "coordinate matching" to measure the parallel amount.

The significance of facts documents to the search query key phrases is being captured by coordinate matching. The essentials are the search facility and privacy protective over encrypted cloud statistics. When we observe at large number of statistics files and records users in the cloud, it is difficult for the necessities of overall performance, usability and scalability. While coming across the facts, the big number of records documents within the cloud server attain to outcome applicable rank in place of returning undistinguishable consequences. To recover the hunt correctness ranking scheme cares more than one keyword search. Present day google network seek devices facts users provide set of key phrases instead of precise keyword search significance to retrieve the most extensive information. Synchronize pairing of query keywords is "Coordinate matching", which can be relevance to that report to the query. The large quantity of files needs the cloud server to perform significance rating as a result, as a substitute returning all result files for effective retrieval of records. Statistics users can discover the most relevant facts using rating gadget in preference to burdensome sorting thru every match inside the facts collection [8]. However, this can bring about a large fee in terms of records, ease of use. For example, the present fashions for keyword- based information retrieval, that are frequently used on the plaintext facts, can't be carried out directly to the encrypted facts. It is impractical to download all information in the cloud and to decrypt regionally. To resolve the above hassle, researchers have some fashionable-cause solutions with in fact homomorphic encryption or blind Random-Access Memory’s [9] built. These techniques aren't practical because of their excessive computational charge for each the cloud Sever and customers. Proposed scheme to reap bendy are in search of for sub-linear trying to find time and cope with the deletion and insertion of files. To decorate the search result, we want inexperienced techniques to perform similarity search over massive number of encrypted records.

2 RELATED STUDY

Searchable encryption schemes permit the customers to hold the encrypted records to the cloud and execute key-word are searching for over cipher textual content vicinity. Due to amazing cryptography primitives, searchable encryption schemes may be built using public key-based totally completely cryptography or symmetric key primarily based sincerely cryptography. These early works are unattached keyword Boolean are searching for schemes, which are pretty easy in terms of capability. Afterward, massive works had been proposed below specific hazard models to benefit various are searching for functionality, alongside unattached keyword are searching for, similarity are looking for, multi-keyword Boolean search, ranked are looking for, and multi keyword Ranked are trying to find and so on. A famous method to shield the records confidentiality is to encrypt the records earlier than outsourcing. Searchable encryption schemes permit the consumer to hold the encrypted statistics to the cloud and execute key-word searching for over cipher text area. So, some distance, sufficient works have been proposed underneath special risk fashions to acquire several looking for functionality, inclusive of single key-phrase are seeking out, similarity search, multi- key-word Boolean attempting to find, ranked seek, multi- key-phrase ranked are seeking out, and so on. Among them, multi-key-word ranked are seeking achieves more and more interest for its sensible applicability. Recently, some dynamic schemes have been proposed to help placing and deleting operations on record series.

Many looking techniques over encrypted cloud records have proposed. S.Deshpande [11] recommended a way looking over encrypted cloud information the usage of fuzzy key phrases. They used Edit distance to quantify key-phrase similarity and advanced two techniques on building fuzzy key phrase devices to reap optimized storage and example overheads. Cong wang et al. [12] Has proposed a manner ranked keyword are seeking over encrypted cloud information the use of key-word frequency and order maintaining encryption. It allows best unattached key phrases at a time. Is the important thing-word frequency identifying file document score. Rank given to every file based totally on the relevance rating of that file. Top ranked documents have despatched to customers as a substitute all documents. To decorate search functionality N. Cao et al. [13] Have proposed a scheme helping conjunctive key phrases are looking for. It is privacy – maintaining multi-keyword ranked are seeking approach the usage of symmetric encryption. M. Chou et al. [14] proposed an answer for fuzzy multi-key-word search over encrypted cloud facts the usage of privacy aware B-Tree. They used a co-incidence possibility method to find out beneficial multi-key phrases for publishing information, files and applicable fuzzy key-phrase gadgets constructed the use of edit distance. The scheme in (Zhiyong et al., 2013) is used to remedy the hassle of dynamic updating of key-phrase, suggest a modern set of guidelines to generate trapdoor, and reduce the effect of the digital word on the end quit end result rating. In the (Orencik et al., 2013), they proposed a multi-key ciphertext retrieval scheme. They use the vector place version to represent file index and question requests. The key- word weights are brought into index and query vector in (Sun et al., 2013). (Yu et al., 2013) located a vector location version and homomorphic encryption scheme to provide higher seek accuracy.

2.1 Drawbacks

1.All these multi-keywords seek schemes retrieve seek consequences primarily based at the life of key phrases, which cannot provide ideal stop end result rating capability.

2.Some early Works have found out the ranked searching for the usage of order- retaining techniques, but they're designed only for single keyword seek.

3.Huge value in terms of records usability. For instance, the triumphing techniques on key-phrase-based totally information retrieval, which might be broadly used at the plaintext data, can't be without delay completed at the encrypted records. Downloading all the facts from the cloud and decrypt regionally is glaringly impractical.

3 PROPOSED SYSTEM

3.1 The key contributions of our art work can be summarized as follows:

(1) We use Intelligent Multi-Keyword Ranked Frequency Pattern (IMKRFP) to encrypt data. The algorithm can lessen the computational overhead of the encryption and decryption. And maximum essential of all, it may however calculate correlation and are looking for cease result ranking within the ciphertext.

(2) We recommend a multi-key-word rating seek scheme primarily based on semantic extension. With semantic extensions to appearance key phrases, you can help particular, complete searches without requiring decryption. It lets in the cloud server to decide whether or not or no longer or now not a given document includes unique keywords or related data. It does no longer need to recognize any- factor about key phrases and files.

4 PROBLEM STATEMENT

Actually, large extensive type of on-call for information customers and huge number of statistics documents inside the cloud, this issue is hard. It is essential for the quest facility to permit multi keyword search query and make to be had quit end result comparison ranking to peer the effective information retrieval requirement. To increase the quest result accuracy as well as to supplement the consumer searching revel in, it's also crucial for such rating device to assist multiple key phrases search, as unattached key-word are seeking frequently yields excessive coarse results. The searchable encryption approach helps to give encrypted information as documents and is of the equal opinion a user to firmly are trying to find over single key-word and retrieve files of trouble.

4.1 SYSTEM MODEL

The maximum crucial images are to establish the proper machine version when we're designing the multi- key-word ranking seek scheme. The version out- property the way of encrypting, importing, and retrieving ciphertext. To greater as it ought to be querying the target document, we regard the cloud garage tool as a model which has three entities. There are information proprietor, legal person, and Cloud provider organization (CSP). The talents of the 3 entities are verified in Figure 1.

Table-1: Notations and Preliminaries.

1. F -The outsourced facts textual content report series can be represented as
2. F= {f1; f2; …... Fn} n is the variety of documents, and fi represents a separate document.
3. W -Keywords that extracted from outsourcing data
4. W= {w1; w2; ... wn}, m is the quantity of key phrases.
5. Wq -A subset of key phrases that represents the key used as a query.
6. I -Index built through keywords.
7. TW -Trapdoor generated by using seek keyword w.
8. SW -Keywords extended set of W
9. SW= {S1, S2……Sn}

C -Encrypted outsourced information textual content document series may be represented as C= {C1; C2;….. Cn} n is the kind of files, and CI represents a separate encrypted document. Data owner, the patron who outsource the statistics to the cloud may be a character or a commercial enterprise business enterprise. They outsource the documents using ciphertexts. The record's collections are represented thru F={F1; F2;.......Fn}. The ciphertext's collections are represented by means of C= {C1; C2; .... Cn}. They set up the index i consistent with the important thing phrases extracted from the text and then they upload encrypted records and indexes to the cloud server.

Authorized person, clients are allowed to retrieve cloud statistics. Authorized customers use the trapdoor to send retrieval requests to CSP. And legal customers decrypt the outcomes retrieved from the cloud server into plaintext. Cloud carrier businesses, CSP in particular responsible for storing client's facts, looking the ciphertext for authorized customers and extending the semantic statistics of the important thing terms which can be from the prison customers. And then get the matching documents from the searchable index. Data decryption is not finished at a few stages within the hunt. Here is the retrieval manner.

1) In this machine model, anticipate that the facts proprietor holds a group of files, that is F={f1; f2; ………. Fn}. And they may assemble the index i ordinary with the vital aspect-word extracted from the document's series. After facts proprietor encrypts the index and report's collection, they will add ciphertexts to the cloud server.

2) Authorized customers use queried key terms wq and key supplied with the beneficial resource of using statistics proprietor to calculate the trapdoor TW and the supply a are trying to find request to CSP.

3) The cloud server gets key-word's set SW through extending semantics of the obtained gate. After looking searchable index, CSP will move again an encrypted file L0 € C which rank consistent with the applicable degree.

4) Authorized clients use the vital thing to decrypt ciphertext L0 which retrieved from server to collect the plaintext L. The secret is secure via way of the records proprietor.

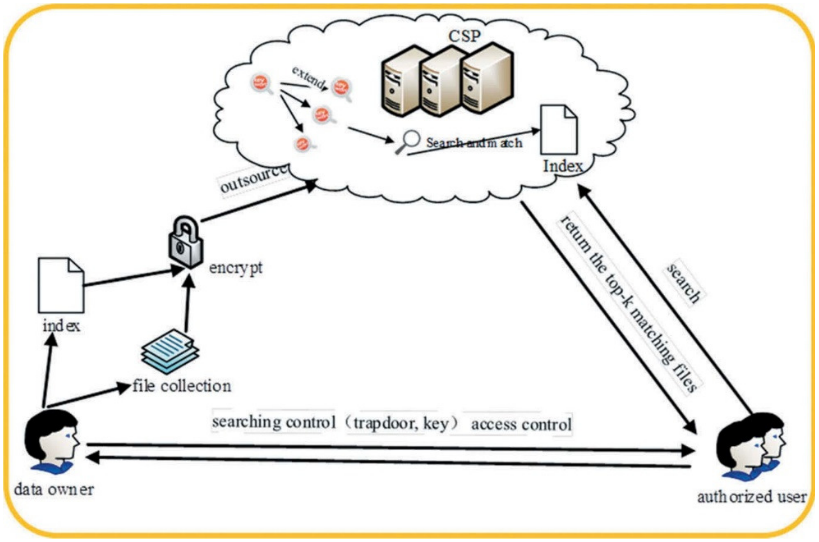

Figure-1: System Model

5. SEARCHABLE ENCRYPTION BASED MOSTLY ON SEMANTIC EXTENSION

The first Symmetrical Searchable Encryption (SSE) scheme and the quest of the scheme is linear within the size of the records collection. Proposed formal security definitions for SSE and advanced a gadget based on Bloom filter. It is proposed that two structures (SSE -1 and a pair of) that the optimal seek time is reached. Your SSE 1 scheme is at ease in opposition to attacks Chosen- Keyword (CKA1) and SSE -2 is at ease towards adaptive chosen- key-word assaults (CKA2). These early works are single key-word Boolean search schemes that are quite simple in phrases of capability.

5.1 Scheme Overview

In order to higher defend the privateness of out- sourced statistics and decrease conversation over- head, we advise a multi-key-word rating Searchable Encryption Scheme Based on Semantic Extension (SESE). First, we used the Term Frequency–Inverse Document Frequency (TF- IDF) approach to extract key phrases from the out- sourced records and assemble indexes. This method can build the index greater correctly. Data proprietor makes use of Intelligent Multi-Keyword Ranked Frequency Pattern (IMKRFP) to encrypt the outsourcing information and add records to the cloud server. When a certified person sends a are seeking for request to the cloud server, the cloud server does no longer at once decrypt the outsourced data. The cloud server makes use of the FP-Growth set of rules to semantically expand key terms and retrieve comparable intention documents. The pinnacle is trying to find results are however to the legal patron. The scheme includes education phase and retrieval phase.

5.2 Preparation phase

The coaching section takes place earlier than the fact's owner outsources the statistics to the cloud server. It consists of growing indexes and encrypting information. The data proprietor creates the index first. Then, the index is despatched to cloud server along with the encrypted facts file. In order to as it ought to be retrieving the desired documents, the index need to incorporate sufficient data. However, the index cannot show any unique information.

5.3 Create Index

5.3.1 Index table Structure: We collect the index in the hash table using the inverted index approach. The inverted index is specially composed of key phrases desk and file list desk. Keywords table is specifically used to preserve keywords. The document column desk is in most cases used to save files composed of key terms and the score of keywords in that file. It is critical to phrase that a key-word can also moreover exist in multiple documents. The inverted index table is established in Figure 2.

5.3.2 The step of creating index.

Procedure: Build index

```
Step-1: scan file's collections
Step-2: F= {f1; f2; …….fn};
Step-3: stop filtering word, delete punctuation and other unimportant information;
Step-4: extract keywords from F to form a set w= {w1; w2; ....wm}; and build index i;
Step-5: insert the file's id which is corresponding to keywords into indexical file list;
Step-6: for (i=1, i<=m, i++)
Step-7: for (j=1, j<=n, j++)
Step-8: if (keyword wi € file fj)
Step-9: calculate the weight of wi in file fj denoted as score;
Step-10: insert the score into indexical filelist;
Step-11: End if
Step-12: end for
Step-13: end for
Step-14: return i
```

Where the key-word's weight is calculated as follows:

TF-IDF is a keyword's weight calculation technique. We use it to calculate the rating of key- phrases inside the file. The traditional TF-IDF algorithm increases the place weight to improve the accuracy of key-word extraction with a view to keep away from the semantic deviation in the next semantic extension.

The score of key-word is calculated in Equation (1):

$$\mathbf{score=Ni*log(n/ni+\beta)\ *mi\times lwi/L} \text{ -------------------------- } \mathbf{(1)}$$

Ni indicates the frequency of keywords wi in a file fj. n represents the overall variety of files. Ni represents the general style of key phrases wi appearing within the record. β represents an empirical value. Generally, take 0.01, 0.1, and 1. lwi represents the paragraph number wherein key-phrase is acting within the document. L is the total quantity of paragraphs. Mi is the region weight of keyword within the record.

Keyword is classified as global or close by key-word. It can be seen from the distribution of key phrases inside the paragraph. The more paragraphs the keyword is in the greater common the key-phrase is, and the better score is.

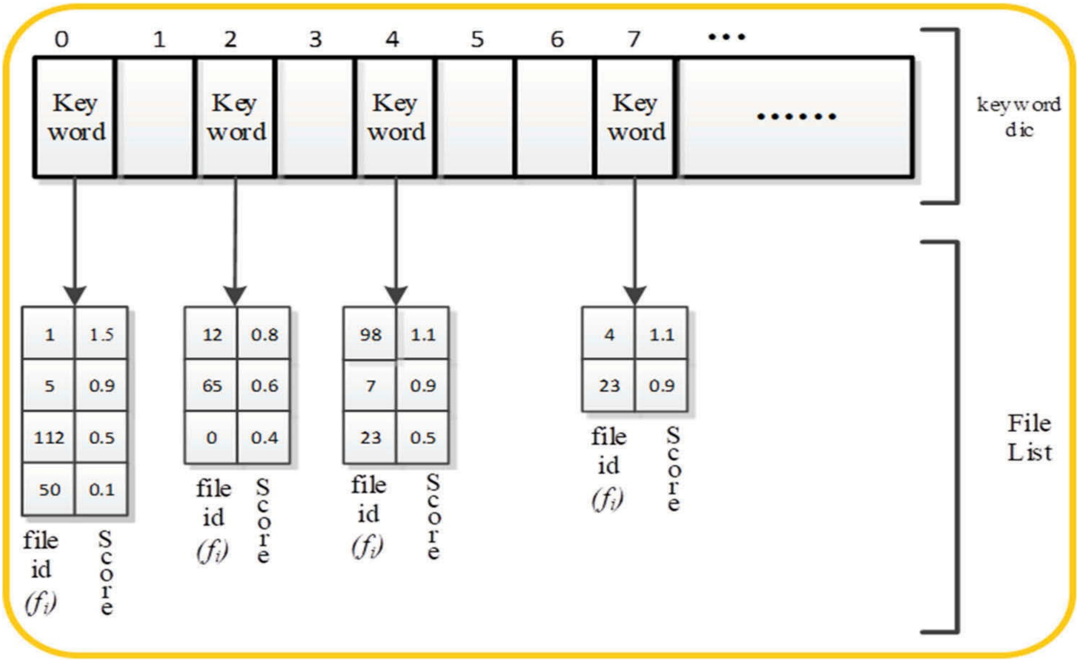

Figure-6.2: Inverted Index Table.

5.3.3 Encrypt records

In order to better shield the privacy of the out- sourced statistics, we use the ECC set of policies primarily based totally on bilinear mapping to encrypt the report. Compared with the homomorphic encryption set of suggestions, this set of pointers has much less computation and quicker encryption and decryption.

5.3.4 Key generation

Common mild-weight solutions embody PRESENT and RC4 set of rules. The benefits of mild-weight answers are that they have got short period of key, smooth form, and espresso useful resource intake. Compared with these two answers, the huge benefit of ECC is that it's miles more at ease and greater hard to crack. Outsourcing facts encryption is the maximum important part of the searchable encryption scheme proposed on this paper. Data proprietors need quick encrypted instances. But they could pick out their outsourced records greater comfy. Therefore, in this summary we pick out ECC.

Intelligent Multi-Keyword Ranked Frequency Pattern (IMKRFP) is a combination of ECC and DSA. The complete signature technique is much like DSA, besides that the set of rules followed within the signature is ECC, and the final signature price is likewise divided in to r and s. The non-singular elliptic curve in actual range place is Ep.

It is tested in Equation (2).

$$\mathbf{Y^2= x^3+ax+b\ (mod\ p)\ ------------------------\ (2)}$$

where p is a prime. Kp is a large prime field. a, b and p are the element of Kp, and the elliptic curve meets 4a3+27b2! =0. We use E_P (a, b) represents the elliptic curve. We take a random point G on the elliptic curve E_P as the base point. Select two integers r, s (0 < r, s < p) as the private key of the data owner. The private key multiplies with G. And then we will get the public key S, R.

5.3.5 Data Encryption:

Subsequently the index is shaped and the key is generated, the collections of files are encrypted using the Intelligent Multi-Keyword Ranked Frequency Pattern (IMKRFP) to make convinced the confidentiality of the uploaded facts. The encrypted procedure is as follows. Let the record be signified by using F. F= {m1; m2; ... Mn}; Where, the period of F is n. And mi is a binary string and index.

The cipher- text is intended as confirmed in Equation (3).

$$C_i = m_i + r * PK_{server}\text{--------------------------------}(3)\ ,$$

in which Ci is the ciphertext by using encrypting the factor G. PKserver is the general public key of cloud server. Encryption set of rules is as follows.

5.3.5.1 Procedure: Encrypt Information

Step-1: Choose a random point G on the elliptic curve E_P;

Step-2: Pick two integers r; s (0 < r; s < p) because the non-public key (private key);

Step-3: Calculate PKowner=r *G, PK_{server}=s*G;

Step-4: Public key of data owner and cloud server is PKowner = r * G, PKserver= s* G;

Step-5: for (i =1, i ≤ n, i ++)

Step-6: Ci = mi + r * P_{KS}

Step-7: end for

Step-8: send the collections of ciphertext C to the cloud server.

$$C = \{C1, C2, \text{---------------}Cn\};$$

Data owners upload collections of encrypted files and comfortable searchable indexes to the server. This index offers enough statistics for crook customers. You can seek without gifting away any purchaser's information. This approach can save you key-word guessing assaults and offer controllable queries for felonious clients.

6. F-P Growth Algorithm

a) FP Growth Algorithm is abbreviated as Frequent Pattern Growth set of policies. It is an enhancement of Apriori set of guidelines in Association Rule Learning. FP increase set of rules is used for discovering common itemset in a transaction database with none era of applicants. FP boom represents common gadgets in common pattern bushes which also may be referred to as a FP-tree.

b) A common pattern is generated wherein the candidate generation isn't needed. FP increase set of regulations represents the database in a tree pattern known as frequent sample tree or FP tree. FP Tree is a tree-like shape that's made with the initial item devices of the database. The cause of the FP tree is to discover the maximum common sample wherein each node of the FP tree represents an object of the itemset.

c) This tree structure will hold the association or relation among the object gadgets. The database is separated through the use of one common item. This fragmented or separated element is called as sample fragment. The object units of those fragmented styles are studied. Hence, the search for common item gadgets is decreased comparatively.

d) In FP Tree, the basis node represents null even as the decrease nodes constitute the object devices in Database. The relation of the nodes with the decrease nodes that is the object gadgets with the opposite item devices are maintained at the equal time as forming the tree.

6.1 Frequent Pattern Algorithm Steps

The frequent pattern boom set of guidelines allows us to discover the common pattern without the technology of candidates.

Let us see the stairs accompanied inside the commonplace sample through using commonplace sample growth set of policies:

Step 1: The first step is to scan the complete database to discover the feasible occurrences of the object units in the database. This step is the similar to the first step of Apriori algorithm. Number of 1-itemsets in the database is called aid consider or frequency of 1-itemset.

Step 2: The 2nd step in the FP increase set of guidelines, is to construct the FP tree. Create the foundation of the tree where the basis is represented with the aid of manner of null.

Step 3: The next step is to check the database over again and observe the transactions. Examine the number one transaction and discover the itemset within the database. The itemset with the maximum depend is taken on the pinnacle and the itemset with decrease count is taken at bottom and so forth. Which approach that the branch of the tree is built with transaction item units in descending order of rely.

Step 4: The next step is to take a look at the transaction in the database. The item gadgets are taken care of in descending order of rely. If any itemset of this transaction is already found in every different department, then this transaction branch can also percent a common prefix to the basis of the FP Growth set of policies. This way that the not unusual itemset is hooked up to the brand-new node of every different itemset on this transaction.

Step 5: Next step is that the keep in mind of the itemset is prolonged as it takes vicinity in the transactions. Both the common node and new node keep in mind is incremented with the aid of way of 1 as they'll be created and connected consistent with transactions.

Step 6: This step is achieved to mine the FP Tree that's created. In this step, the bottom node is examined first along aspect the connections of the lowest nodes. The lowest node represents the frequency sample of period 1. Traverse the path within the FP Tree. These paths are referred to as a conditional pattern base. Conditional sample base is a sub-database consists of prefix paths in the FP tree with the lowest node as suffix.

Step 7: Next step is to collect a Conditional FP Tree, that is fashioned by means of the use of a be counted quantity of object units in the direction. The item units which satisfy the threshold manual is taken into consideration inside the Conditional FP Tree.

Step 8: Final step is to generate Frequent Patterns from the conditional FP Tree.

Advantages

1. FP set of rules is Faster than Apriori set of regulations.
2. Candidate generation isn't always needed.
3. There could be first-class passes over dataset.

Disadvantages

1. FP tree may not be capable of fit in memory.
2. FP tree is luxurious to build the set of rules.

CSP extends keywords by manner of the FP-Growth set of rules and get key-word's set Sw and question index. To avoid immoderate extension key phrases in the record we want to look at the complete correlation Cscore amongst extended key phrases and documents. The score represents the relevance among key-phrase w and file. The Ri represents the semantic relevance's score among searchable key terms and key- phrases.

Cscore is calculated as shown in Equation (4).

$$\mathbf{C_{score}=score_{w}+\sum_{si\text{€}sw}^{scoresi} X\ Ri\text{----------------------------}(4)}$$

By setting the threshold price, we're able to calculate the entire correlation fee many of the record and the user's query. Finally, the quest effects will include semantically applicable key phrases. The cloud server returns top-ok documents. The results of this question may be greater consistent with user requirements.

6.2 Ciphertext retrieval

During the retrieval segment, the criminal clients use trapdoor on retrieve files saved in the cloud server. The trapdoor Twi of the key-phrase is calculated the use of the statistics owner's personal key PRs.

The calculate approach of trapdoor is confirmed in Equation (5).

$$\mathbf{Twi} = \sum_{i=1}^{m} \boldsymbol{H}(\boldsymbol{wi})^{r} \text{---}\mathbf{(5)}$$

Where m is the number of key phrases and H is a hash collision feature SHA-1.

CSP extends the vital element-word to get the dataset Sw. After trapdoor processing, it's far dispatched to CSP to suit the index. Then, the top-excellent sufficient ciphertexts matching the index are another time to the confinement consumers. Otherwise, the ciphertext series does not encompass the files to be searched thru the felony clients.

The FP-tree is mined as follows. Start from each frequent length-1 pattern (as an initial Suffix Pattern), construct its conditional pattern base (a "sub-database", which consists of the set of prefix paths in the FP-tree cp-occurring with the suffix pattern), then construct its(conditional) FP-tree, and perform recursively on the tree. The pattern growth is achieved by the concatenation of the suffix pattern with the frequent patterns generated from a conditional FP-tree. The FP-Growth algorithm to find frequent Itemset in a transaction database without generating candidates. The FP-Growth algorithm represents the database as a tree known as a frequent pattern tree or FP tree. The relationship between the item sets will be maintained by this tree structure.

The FP-tree is summarized in following table and detailed as follows. We first consider I5, which is the last item in L, rather than the first. The reason for starting at the end of the list will become apparent as we explain the FP-tree process. I5 occurs in two FP tree branches of figure 6.7 (The occurrences of I5 can easily be found by following its chain of node-links.) The paths formed by these branches are {I2, I1, I5:1} and {I2, I1, I3, I5:1}. Therefore, considering I5 as a suffix, its corresponding two prefix paths are {I2, I1:1} and {I2, I1, I3:1}, which form its conditional pattern base as a transaction database, we build a 15-conditional FP-tree, which contains a single path, {I2:2, I1:2}; I3 is not included because its support count of 1 is less than the minimum support count. The single path generates all the combinations of frequent patterns: {I2, I5:2}, {I1, I5:2}, {I2, I1, I5:2}. For I4, its two prefix paths form the conditional pattern base, {{I2, I1:1}, {I2:I1}}, which generates a single-node conditional FP-tree, {I1:2}, and derives one frequent pattern, {I2, I4:2}.

Table-2: The FP-Tree by Creating Conditional (Sub-)Pattern Bases

Item	Conditional Pattern Base	Conditional FP-tree	Frequent Patterns Generated
I5	{{I2, I1: 1}, {I2, I1, I3:1}}	(I2: 2, I1: 2)	{I2, I5: 2},{I1, I5: 2}, {I2, I1, I5: 2}
I4	{{I2,I1:1},{I2:1}	(I2:2)	{I2,I4:2}
I3	{{I2,I1:2},{I2:2},{I1:2}}	(I2:4,I1:2), (I1:2)	{I2,I3:4},{I1,I3:4},{I2,I1,I3:2}
I1	{{I2:4}	(I2:4)	{I2, I1:4}

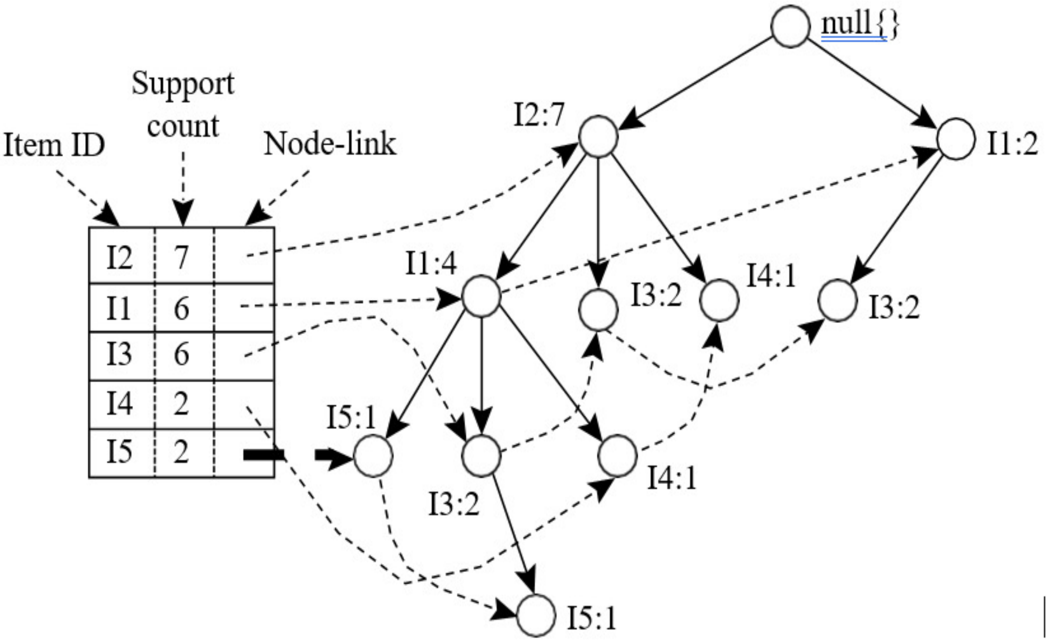

Figure-3: An FP-tree registers compressed, frequent pattern information

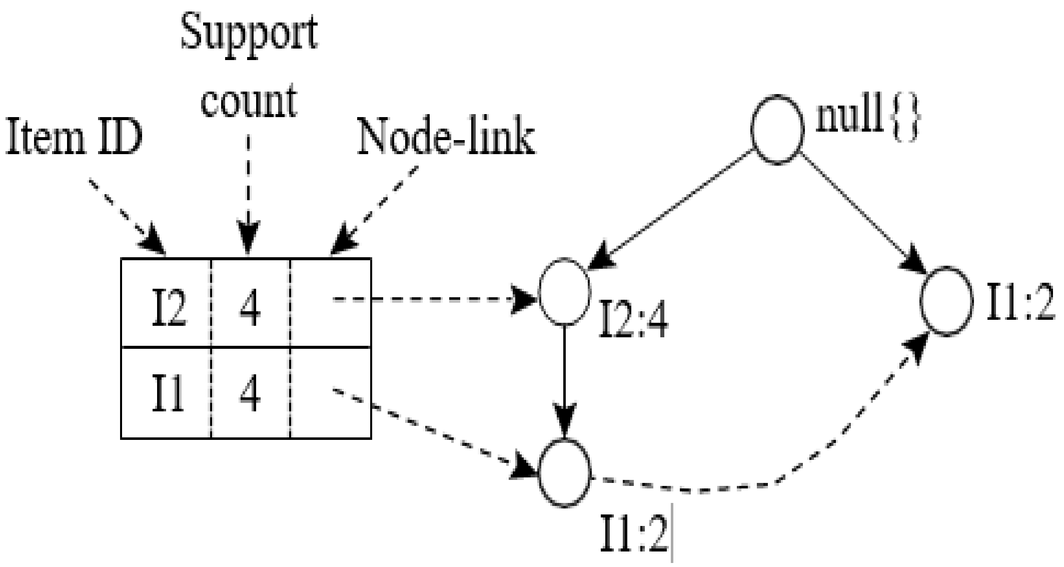

Figure-4: The conditional FP-tree associated with the conditional node I3.

Similar to the preceding analysis, I3's conditional pattern base is {{I2, I1:2}, {I2:2},{I1:2}}. Its conditional FP-tree has two branches, (I2: 4, I1:2) and (I1:2) as shown figure 6.8, which generates the set of patterns {{I2, I3:4}, {I1, I3:4}, {I2, I1, I3:2}}. Finally, I1's conditional pattern base is {{I2:4}}, with an FP-tree that contains only one node, (I2:4), which generates one frequent pattern, {{I2, I1:4}. This process is summarized in following FP-Growth Algorithm.

7. Algorithm: FP-Growth. Mine frequent itemsets using an FP-tree by pattern fragment growth.

Input:
D, a transaction database; min_sup, the minimum support count threshold.

Output: The complete set of frequent patterns.

Method:

1. The FP-tree is constructed in the following steps:

(a) Scan the transaction database D once. Collect F, the set of frequent items, and their support counts. Sort F in support count descending order as L, the list of frequent items.

(b) Create the root of an FP-tree, and label it as "null". For each transaction Trans in D do the following.

Select and sort the frequent items in Trans according to the order of L. Let the sorted frequent item list in Trans be [p|P], where p is the first element and P is the remaining list.

Call insert_tree([p|P], T), which is performed as follows.

If T has a child N such that N.item-name=p.item-name, then increment N’s count by 1; else create a new node N, and let its count be 1, its parent link be linked to T, and its node-link to the nodes with the same *item-name* via the node-link structure. If P is nonempty, call insert_tree(P, N) recursively.

7.1 The FP-tree is mined by calling FP_Growth (FP_tree, null), which is implemented as follows.

```
Step-1: If Tree contains a single path P then
Step-2: for each combination (denoted as $\beta$) of the nodes in the path P
Step-3: generate pattern $\beta$ U $\alpha$ with support_count=minimum support count of nodes in $\beta$;
Step-4: else for each ai in the header of Tree
            {
Step-5: generate pattern $\beta$=ai U $\alpha$ with support_count=ai.support_count;
Step-6: construct $\beta$’s conditional pattern base and then $\beta$’s conditional FP-tree Tree $\beta$;
Step-7: if Tree $\beta \neq \emptyset$ then
Step-8: call FP_groth(Tree$\beta$, $\beta$);
            }
```

7.2 FP-Growth Retrieval Process of Proposed Scheme

The process of finding data (generally documents) in the form of text that matches the information required from a set of documents stored on a computer is known as information retrieval (IR). Incorrect user requests are a common problem on IRs; this is caused by user weaknesses in representing their needs in the query. Researchers have suggested numerous solutions to address these limitations; in this review, we proposed an approach based on the FP- Growth algorithm for the quest for frequent itemsets. The key to storing keywords is the cloud storage server, which is denoted by the letter K. (keyword0, keyword1, keyword2, and keyword3). Calculate the support for a single itemset by traversing each record and seeing if it includes an itemset.

Procedure: FP-Growth (DB, ξ)

```
Step 1: Define and clear F-List : F[];
Step 2: foreach T transaction Ti in DB do
    Step 3: foreach Item aj in Ti do
        Step 4: F[ai] ++;
    Step 5: end
Step 6: end
Step 7: Sort F [];
Step 8: Define and clear the root of FP-tree: r;
Step 9: foreach T transaction Ti in DB do
    Step 10: Make Ti ordered according to F;
    Step 11: Call Construct T tree (Ti,r);
Step 12: end
Step 13: foreach item $a_i$ in I do
    Step 12: Call Growth (r, $a_i$, ξ);
Step 13: - end
```

This algorithm begins by compressing the input database, resulting in a frequent pattern tree case. The compressed database is then divided into a few conditional databases, each representing a single specific frequent pattern. Finally, mining of each database is done separately. As a result, the search costs are greatly reduced, resulting in strong selectivity.

PRs is the private key supplied via the statistics proprietor. PKserver is most of the people key supplied via CSP. Authorized users use PRs and PKserver to decrypt the retrieved ciphertext. Finally, the target plaintext set L is obtained.

The decryption equation is Ci - s * r * G =M--------------------(6)

We used the Intelligent Multi-Keyword Ranked Frequency Pattern (IMKRFP) to encrypt files. The set of regulations improves the rate of encryption and decryption and guarantees the security. By extending key phrases, the hazard of the customer to retrieve the reason files is improved.

8 Experiment Results

8.1 Experimental surroundings

SESE scheme uses C++ language and is tested inside the processor is Intel(R) Core (TM)i5-2430 M CPU@2.40 GHz, (Max Turbo Frequency 3.00 GHz; Intel® Turbo Boost Technology 2.0 Frequency‡ 3.00 GHz; Processor Base Frequency 2.40 GHz; Cache 3 MB Intel® Smart Cache), 64-bit windows7 surroundings. The test files' collections are Request for Comments (RFC) database, which includes more than 6000 files.

9 Analysis of experimental results

9.1 Preparation Phase

In the coaching phase, the facts owners are accountable for growing index and encrypting information. By scanning each record and extracting the important thing- phrase we can create index. Then, we file paragraphs quantity in each file in an effort to correctly determine the key phrases. Intelligent Multi-Keyword Ranked Frequency Pattern (IMKRFP) is used to encrypt statistics due to its higher security, faster processing velocity, and smaller garage area. There are the results of experiments.

Figure-5: suggests the encrypted time of documents. When we use SESE to encrypt files, the time is smaller than the TRSE.

	TRSE	SESE	Proposed Method
time(ms)	400	100	50
	500	200	150
	900	300	250
	1100	400	350
	1700	500	450
The size of files(GB)	1	1	1
	2	2	2
	3	3	3
	4	4	4
	5	5	5

Table-3: The values of encrypted time of files.

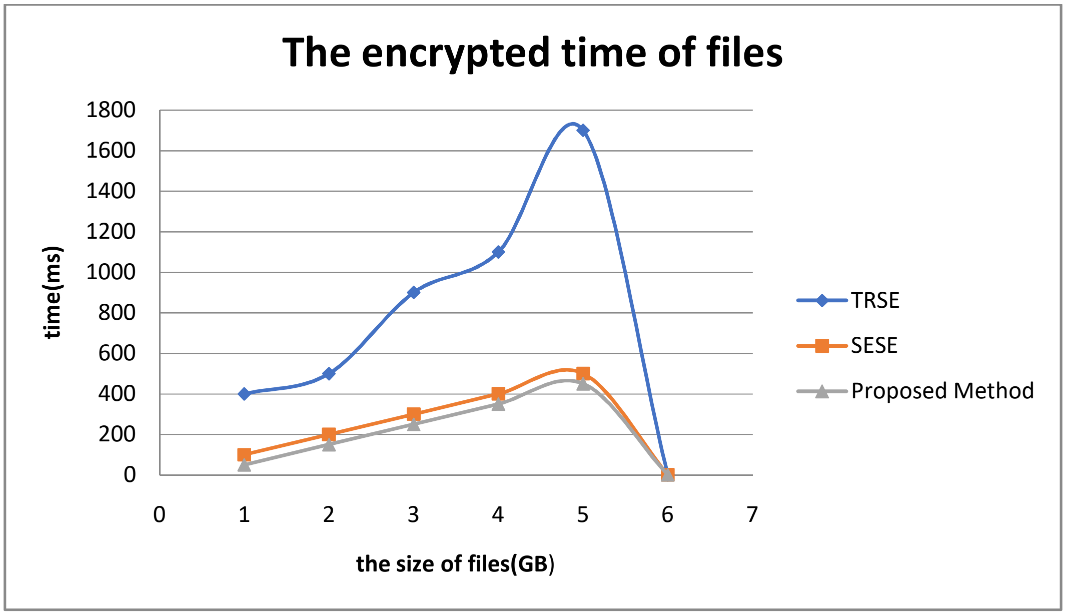

Figure-5: The encrypted time of files

Figure-6: shows the time of creating index in a specific wide variety of files. We need to scan all.

	TRSE	SESE	Proposed Method
time(ms)	30	30	20
	50	45	40
	70	65	60
	90	80	70
	110	100	90
The size of files (GB)	1	1	1
	2	2	2
	3	3	3
	4	4	4
	5	5	5

Table-4: The values of time to build index in different files's numbers

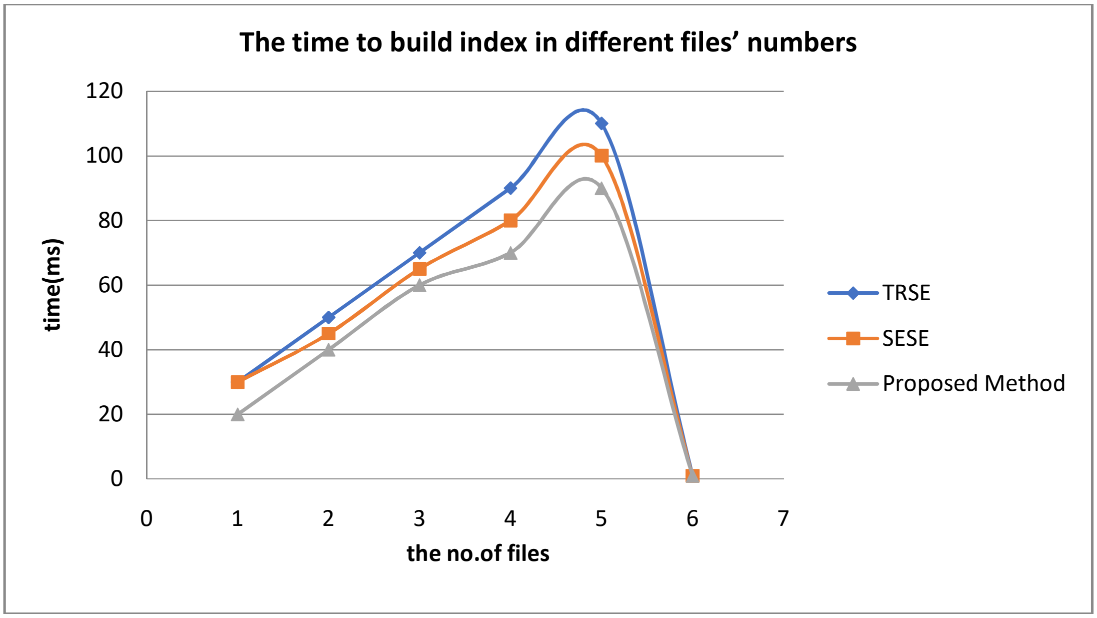

Figure-6: The time to build index in different files numbers

Figure-7 shows that the time of creating semantic relation database is proportional to the number of files. We use the association rule mining algorithm to create the semantic relational database.By comparing with the TRSE scheme, it is proved that the performance of index creation and data's encryption in SESE is feasible.

	TRSE	SESE	Proposed Method
time(ms)	10	10	10
	20	15	15
	40	20	16
	80	50	40
	110	60	50
	140	70	60
	180	80	70
	210	90	80
The no.of keywords	0	0	0
	500	500	500
	1000	1000	1000
	1500	1500	1500
	2000	2000	2000
	2500	2500	2500
	3000	3000	3000
	3500	3500	3500

Table-5: The value of generate trapdoor in different number of keywords

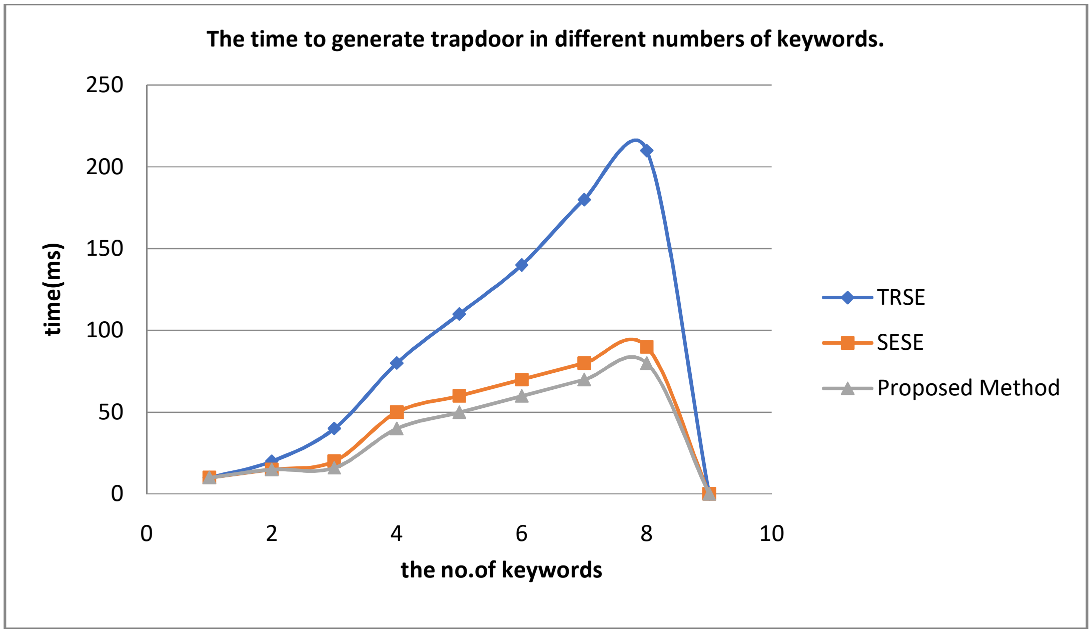

Figure-7: The time to generate trapdoor in different numbers of keywords

The files to extract the keywords' set. And build an inverted index including the relevant values.

Now we are capable of have a look at our scheme's computational complexity. As we recounted, we use Intelligent Multi-Keyword Ranked Frequency Pattern (IMKRFP) to encrypt files as it has lots plenty less encrypted time. Compared with RSA (1024), the encrypted time of IMKRFP (163) just wishes 3.0 ms. Meanwhile, the time required to interrupt ECC with the handiest assault technique is exponential. T(n)=O (exp (ln n)/p (max)).P(max) is the maximum critical pinnacle divisor. Assuming the facts searched with the aid of way of manner of someone includes n key terms. To enumerate all of the related combinations with the aid of exhaustive approach, the type of operations required is 2^n.

At this time, extended key terms will multiply. T(n)=o(m2^n). However, a number of the extended key terms want to be duplicated, which ends up in the waste of computation time and garage place. We can reduce the time complexity to O ($\sqrt{2^n}$)as a minimum by way of way of the use of FP-Growth. Dramatically reduce the computational complexity. All in all, our scheme quick encrypted time and immoderate protection ordinary overall performance.

9. Conclusions

In our scheme, we use Intelligent Multi-Keyword Ranked Frequency Pattern (IMKRFP) to encrypt the collection of files. The algorithm can reduce the computation cost of encryption and decryption. And upload encrypted files and indexes to the cloud server. The authorized users use the hashcollision function to generate the trapdoor and send it to CSP searching the matching ciphertext. CSP uses FP-Growth algorithm to extend keywords and search index to match the ciphertext. Because searchable encryption is now primarilyused for accurate retrieval, some files containing semantically related words are easily ignored. Therefore, our scheme improves the recall rate of files. And this scheme can not only improve the speed of search, but also protect the privacy of users from being acquired by CSP. Improve the confidentiality of index and content, keyword confidentiality. However, our method also has some disadvantages when compared with the real ones running in current cloud systems. Content- awareness encryption (CAE) is a common encrypted method in cloud systems. It isa method of encryption based on policy Settings. Compare with CAE, our method is not flexible. Alldata are encrypted in a uniform method but there are various formats of data in the cloud systems. So, in future time, we will further study how to make Intelligent Multi-Keyword Ranked Frequency Pattern (IMKRFP) more flexible.

REFERENCES

[1] Bag, A., Patranabis, S., & Tribhuvan, L. (2018). Hardware acceleration of searchable encryption, Conference on computer and communications security (pp. 2201–2203). Toronto, Canada: ACM. https://doi.org/10.1145/3243734.3278509.

[2] Bo, Z., & Fangguo, Z. (2011). An efficient public key encryption with conjunctive-subset keywords search. Journal of Network and Computer Applications, 34(1), 262–267. https://doi.org/10.1016/j.jnca.2010.07.007

[3] Boneh, D., Crescenzo, G. D., Ostrovsky, R., & Persiano, G. (2004). Public key encryption with keyword search. advances in cryptology - EUROCRYPT 2004, International conference on the theory and applications of cryptographic techniques, Interlaken, Switzerland, May 2- 6, 2004, Proceedings.

[4] Springer-Verlag. https://doi.org/10.1007/978-3-540-24676- 3_30. Boneh, D., & Waters, B. (2006). Conjunctive, subset, and range queries on encrypted data. Proceedings of the 4th conference on Theory of cryptography. Berlin, Heidelberg: Springer-Verlag. https://doi.org/10.1007/978-3-540-70936- 7_29.

[5] Byun, J. W., Lee, D. H., & Lim, J. (2006). Efficient conjunctive keyword search on encrypted data storage system. European Conference on Public Key Infrastructure: Theory & Practice. Turin, Italy: Springer-Verlag. https://doi.org/ 10.1007/11774716_15.

[6] Chengyu, H., & Pengtao, L. (2011). Decryptable searchable encryption with a designated tester. Procedia Engineering, 15(none), 1737–1741. https://doi.org/10.1016/j.proeng. 2011.08.324 Dwan, X., Song, D., & Wagner, A. P. (2002). Practical techniques for searches on encrypted data. Proc. of the 2000 IEEE security and privacy symposium, May. DC, United States: IEEE. https://doi.org/10.1109/SECPRI.2000.848445.

[7] Etemad, M., Küpçü, A., Papamanthou, C., & Evans, D. (2018). Efficient dynamic searchable encryption with forward privacy. Proceedings on Privacy Enhancing Technologies, 2018(1), 5–20. https://doi.org/10.1515/ popets-2018-0002.

[8] Fu, Z., Wu, X., Wang, Q., & Ren, K. (2017). Enabling central keyword-based semantic extension search over encrypted outsourced data. IEEE Transactions on Information Forensics & Security, 12(12), 2986–2997. https://doi.org/ 10.1109/TIFS.2017.2730365.

[9] Golle, P., Staddon, J., & Waters, B. R. (2004). Secure conjunctive keyword search over encrypted data. Applied cryptography and network security, Second international conference, ACNS 2004, Yellow Mountain, China, June 8-11, 2004, Proceedings. https://doi.org/10.1007/978- 3-540-24852-1_3.

[10] Hoang, T., Yavuz, A. A., & Guajardo, J. (2016). Practical and secure dynamic searchable encryption via oblivious access on distributed data structure. Conference on computer security applications. California, USA: ACM, 302–313. https://doi.org/10.1145/2991079.2991088.

[11] Hoang, T., Yavuz, A. A., & Merchan, J. G. (2019). A secure searchable encryption framework for privacy-critical cloud storage services. IEEE Transactions on Services Computing, PP(99), 1. https://doi.org/10.1109/TSC.2019.2897096

[12] Ibraimi, L., Nikova, S., Hartel, P. H., & Jonker, W. (2011). Publickey encryption with delegated search. applied cryptography and network security. 9th international conference, ACNS 2011, Nerja, Spain, June 7-10, 2011. Proceedings. Springer Verlag. https://doi.org/10.1007/978-3-642-21554-4_31.

[13] Jiguo, L., Yuerong, S., & Yichen, Z. (2017). Searchable ciphertext-policy attribute-based encryption with revocation in cloud storage. International Journal of Communication Systems, 30(1), 2942–2942. https://doi. org/10.1002/dac.2942.

[14] Leontiadis, I., & Li, M. (2018). Storage efficient substring searchable symmetric encryption. 6th international workshop on security in cloud computing, Incheon, Korea, June 4-8, 2018. 3–13.

[15] Miao, Y., Ma, J., & Liu, Z. (2016). Revocable and anonymous searchable encryption in multi-user setting. Concurrency & Computation Practice & Experience, 28(4), 1204–1218. https://doi.org/10.1002/cpe.3608

[16] Orencik, C., Kantarcioglu, M., & Savas, E. (2013). A practical and secure multi-keyword search method over encrypted cloud data. cloud computing (CLOUD), 2013 IEEE sixth international conference on. DC, United States: IEEE. https://doi.org/10.1109/CLOUD.2013.18.

[17] Park, D., Kim, K., & Lee, P. (2004). Public key encryption with conjunctive field keyword search. International conference on information security applications. Berlin, Heidelberg: Springer-Verlag. https://doi.org/10.1007/978- 3-540-31815-6_7.

[18] Pasupuleti, S. K., Ramalingam, S., & Buyya, R. (2016). An efficient and secure privacy-preserving approach for outsourced data of resource constrained mobile devices in cloud computing. Journal of Network and Computer Applications, 64(C), 12–22. https://doi.org/10.1016/j.jnca.2015.11.023 [19]

[20] Rizomiliotis, P., Molla, E., & Gritzalis, S. (2017). REX: A searchable symmetric encryption scheme supporting range queries. On cloud computing security workshop. New York, United States: ACM. https://doi.org/10.1145/ 3140649.3140653.

[21] Schwarz, T., Tsui, P., & Litwin, W. (2006). An encrypted, content searchable scalable distributed data structure. International conference on data engineering workshops. DC, United States: IEEE Computer Society. https://doi. org/10.1109/ICDEW.2006.27.

[22] Shangping, W., Shasha, J., & Yaling, Z. S. (2019). Verifiable and multi-keyword searchable attribute-based encryption scheme for cloud storage. IEEE Access, 7(none), 50136–50147. https://doi.org/10.1109/ACCESS.2019.2910828

[23] Sun, W., Wang, B., Cao, N., Li, M., & Li, H. (2013). Verifiable privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking. IEEE Parallel and Distributed Technology Systems and Applications, 25(11), 3025–3035. https://doi.org/10.1109/TPDS.2013.282

[24] Wang, T., & Zhao, Y. (2016). Secure dynamic SSE via access indistinguishable storage. The 11th ACM. New York, United States: ACM. https://doi.org/10.1145/2897845.2897884.

[25] Wer, Z., Yaping, L., Sheng, X., Jie, W., & Zhou, S. W. (2016). Privacy preserving ranked multi-keyword search for multiple data owners in cloud computing. IEEE Transactions on Computers, 65(5), 1566–1577. https://doi.org/10.1109/TC. 2015.2448099.

[26] Yang, K., Zhang, K., Jia, X., Hasan, M. A., & Shen, X. (2016). Privacy-preserving attribute-keyword based data publish-subscribe service on cloud platforms. Information Sciences, S0020025516307666, 387(C). https://doi.org/10. 1016/j.ins.2016.09.020.

[27] Yassine, M., Shojafar, M., & Darwish, A. (2019). Cybersecurity and privacy in cyber physical systems. SubjectsComputer Science, Engineering & Technology.

[28] Yu, J., Lu, P., Zhu, Y., Xue, G., & Li, M. (2013). Toward secure multikeyword top-k retrieval over encrypted cloud data. IEEE Transactions on Dependable and Secure Computing, 10(4), 239–250. https://doi.org/10.1109/TDSC.2013.9

[29] Yuan, X., Wang, X., Chu, Y., Wang, C., & Qian, C. (2015). Towards a scalable, private, and searchable key-value store. Communications & network security. Florence, Italy: IEEE. https://doi.org/10.1109/CNS.2015.7346929.

[30] Zahra, P., Mauro, C., & Chia, M. Y. (2018). RARE: Defeating side channels based on data-deduplication in cloud storage. Proceedings of the INFOCOM Workshops CCSNA 2018, Honolulu, HI, USA. https://doi.org/10.1109/ INFCOMW.2018.8406888.

[31] Zhang, R., Xue, R., Yu, T., & Liu, L. (2016). Dynamic and efficient private keyword search over inverted index–based encrypted data. ACM Transactions on Internet Technology, 16(3), 1–20. https://doi.org/10.1145/2940328

[32] Zhao, W., Qiang, L., Zou, H., Zhang, A., & Li, J. (2018). Privacy-preserving and unforgeable searchable encrypted audit logs for cloud storage. 5th IEEE international conference on cyber security and cloud computing (CSCloud)/ 2018 4th IEEE international conference on edge computing and scalable cloud (EdgeCom), Shanghai, China. 29–34. https://doi.org/10.1109/CSCloud/EdgeCom.2018.00015.

[33] Zhao, Z., Lai, J., Susilo, W., Wang, B., Hu, Y., & Guo, F. (2019). Efficient construction for full black-box accountable authority identity-based Encryption. IEEE Access, 7 (none), 25936–25947. https://doi.org/10.1109/ACCESS. 2019.2900065 Zhiyong, X., Kansheng., W., Luixuan., R., Yow, K., & Chengzhong, X. (2013). Efficient multi-keyword ranked query on encrypted data in the cloud. 2012

IEEE 18th international conference on parallel and distributed systems. Singapore, Singapore: IEEE. https://doi.org/10.1109/ ICPADS.2012.42.