International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 11
July 2016

Secrecy Protecting Rated Multi-Keyword Inspection for Various Data Holders in Cloud Computing

Mr. T.SRIKANTH
Asst. Professor
Department of CSE

Ms. B.MANJULA
M.Tech in Computer Science
Department of CSE

Malla Reddy Institute of Technology, Maisammaguda, Dhulapally Post, Secunderabad and Telengana State, India.

Abstract: with the advent of cloud computing, it has emerge as increasingly popular for facts owners to outsource their statistics to public cloud servers while permitting data customers to retrieve this statistics. for privacy concerns, cozy searches over encrypted cloud information has stimulated numerous studies works under the single proprietor version. but, most cloud servers in exercise do not simply serve one owner; alternatively, they support multiple proprietors to percentage the blessings added by way of cloud computing. in this paper, we advise schemes to cope with privacy keeping ranked multi-keyword search in a multi-owner model (prmsm). To permit cloud servers to perform comfortable search without understanding the real data of each keywords and trapdoors, we systematically assemble a unique relaxed search protocol. to rank the quest consequences and keep the privacy of relevance ratings between key phrases and files, we advocate a unique additive order and privateness maintaining feature family. To save you the attackers from eavesdropping mystery keys and pretending to be prison facts users submitting searches, we propose a unique dynamic mystery key technology protocol and a new records user authentication protocol. moreover, prmsm supports green records consumer revocation.

Vast experiments on actual-international datasets affirm the efficacy and performance of prmsm.

Index terms: cloud computing, ranked keyword seek, more than one proprietors, privateness maintaining, dynamic secret key

1 Advent:

Cloud computing is a subversive era that is changing the way it hardware and software program are designed and purchased. As a new model of computing, cloud computing gives abundant benefits which include smooth get right of entry to, decreased expenses, short deployment and bendy useful resource management, and many others. organizations of all sizes can leverage the cloud to boom innovation and collaboration. regardless of the plentiful blessings of cloud computing, for privacy concerns, people and agency users are reluctant to outsource their touchy statistics, which include emails, non-public fitness records and government private files, to the cloud. this is due to the fact once

touchy information are outsourced to a far flung cloud, the corresponding records proprietors lose direct manage of these facts. Cloud service vendors (csps) would promise to ensure proprietors' information protection the use of mechanisms like virtualization and firewalls. but, these mechanisms do now not guard proprietors' facts privacy from the CSP itself, since the CSP possesses full manipulate of cloud hardware, software, and proprietors' statistics. Encryption on sensitive records before outsourcing can maintain facts privacy towards CSP. however, facts encryption makes the conventional statistics utilization carrier based on plaintext key-word seek a completely tough problem. A trivial approach to this hassle is to down load all the encrypted facts and decrypt them domestically. but, this technique is obviously impractical as it will cause a huge quantity of communique overhead. consequently, developing a cozy search provider over encrypted cloud statistics is of paramount importance. cozy search over encrypted statistics has currently attracted the hobby of many researchers. music et al. first define and clear up the problem of cozy search over encrypted records. They endorse the idea of searchable encryption, which is a cryptographic primitive that permits users to carry out a key-word-primarily based seek on an encrypted dataset, simply as on a plaintext dataset. Searchable encryption is further advanced. however, those schemes are worried frequently with single or boolean key-word seek. Extending those strategies for ranked multi keyword seek will incur heavy computation and storage costs. comfortable search over encrypted cloud facts is first described via Wang et al. and in addition developed. Those researches no longer simplest lessen the computation and storage fee for secure key-word seek over encrypted cloud facts, but additionally increase the category of search function, together with cozy ranked multi-keyword seek, fuzzy keyword seek, and similarity seek. but, a majority of these schemes are restrained to the unmarried-proprietor model. As a remember of reality, maximum cloud servers in exercise do not just serve one records proprietor; rather, they regularly help multiple facts owners to proportion the blessings introduced by cloud computing. for instance, to assist the government in creating a first-class rules on health care carrier, or to help medical institutions behavior useful research, some volunteer patients might conform to percentage their health records on the cloud. To keep their privacy, they will encrypt their own fitness data with their mystery keys. on this state of affairs, only the legal organizations can carry out a cozy search over this encrypted facts contributed by way of more than one data owners. this type of non-public health document sharing system, wherein more than one statistics owners are involved, can be located at mymedwall.com. as compared with the single-proprietor scheme, growing a complete-fledged multi-proprietor scheme could have many new challenging troubles. First, within the single owner scheme, the facts owner has to stay on-line to generate trapdoors (encrypted keywords) for information customers. But, whilst a large amount of statistics proprietors are concerned, asking them to stay on line concurrently to generate trapdoors would significantly affect the flexibility and value of the quest system. second, in view that none people might be willing to share our secret keys with others, distinctive facts owners would prefer to apply their

personal mystery keys to encrypt their mystery information. consequently, it's miles very hard to carry out a secure, handy, and efficient search over the data encrypted with special mystery keys. 0.33, when multiple statistics owners are worried, we need to make sure green person enrollment and revocation mechanisms, so that our gadget enjoys notable protection and scalability. on this paper, we advise PRMSM, a privateness maintaining ranked multi-key-word seek protocol in a multi-proprietor cloud model. To permit cloud servers to perform relaxed search with out understanding the actual price of both keywords and trapdoors, we systematically construct a unique relaxed seek protocol. As a result, one of a kind statistics owners use exclusive keys to encrypt their files and keywords. Authenticated statistics users can trouble a question without knowing mystery keys of those one of a kind records proprietors. To rank the quest consequences and hold the privacy of relevance scores between key phrases and files, we suggest a brand new additive order and privateness preserving characteristic own family, which allows the cloud server return the maximum applicable seek consequences to information customers without revealing any sensitive records. To prevent the attackers from eavesdropping secret keys and pretending to be prison records users filing searches, we suggest a novel dynamic secret key era protocol and a brand new facts user authentication protocol. As a end result, attackers who steal the name of the game key and carry out unlawful searches could be without problems detected. Furthermore, whilst we need to revoke a records consumer, PRMSM guarantees green facts user revocation. Widespread experiments on actual-global datasets affirm the efficacy and efficiency of our proposed schemes. the main contributions of this paper are indexed as follows:

• We outline a multi-proprietor model for privateness retaining key-word search over encrypted cloud records.

• We propose an green information person authentication protocol, which now not most effective prevents attackers from eavesdropping secret keys and pretending to be illegal statistics customers appearing searches, however also allows statistics user authentication and revocation.

• We systematically assemble a novel comfortable seek protocol, which not simplest enables the cloud server to perform comfy ranked keyword seek without knowing the real facts of both key phrases and trapdoors, but additionally allows facts owners to encrypt keywords with self-chosen keys and allows authenticated statistics users to question without knowing these keys.

• We advise an Additive Order and privacy keeping feature own family (AOPPF) which lets in statistics owners to protect the privacy of relevance rankings the usage of distinct functions consistent with their preference, at the same time as nevertheless allowing the cloud server to rank the data files appropriately.

• We conduct widespread experiments on actual-world datasets to verify the efficacy and efficiency of our proposed schemes. The rest of this paper is prepared as follows.

Segment 2 formulates the hassle. section three gives the preliminaries. Segment 4 demonstrates a way to

perform user authentication. segment 5 introduces our novel cozy seek protocol. phase 6 defines AOPPF and illustrates a way to use this approach to carry out privacy-keeping ranked search. segment 7 gives protection evaluation. phase 8 demonstrates the efficiency of our proposed scheme. The associated works are reviewed in phase 9. In section 10, we finish the paper.

2 Trouble components:

On this phase, we present a formal description for the goal hassle in this paper. We first outline a system model and a corresponding hazard version. Then we elucidate the layout dreams of our answer scheme and a listing of notations used in later discussions.

System version: In our multi-proprietor and multi-consumer cloud computing version, four entities are involved, as illustrated in Fig. 1; they are records proprietors, the cloud server, management server, and statistics customers. Facts proprietors have a set of files F. To enable green search operations on those documents if you want to be encrypted, facts proprietors first build a comfortable

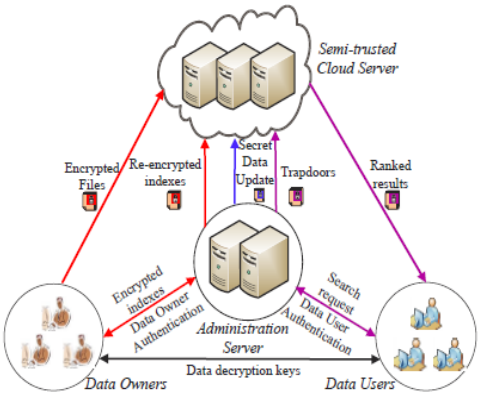


Fig. 1: Architecture of privacy preserving keyword search in a multi-owner and multi-user cloud model

searchable index I on the keyword set W extracted from F, then they post I to the management server. in the end, records proprietors encrypt their documents F and outsource the corresponding encrypted files C to the cloud server. Upon receiving I, the administration server re-encrypts I for the authenticated statistics owners and outsources the re-encrypted index to the cloud server.

Once a records consumer desires to seek t key phrases over these encrypted files saved at the cloud server, he first computes the corresponding trapdoors and submits them to the management server. once the records consumer is authenticated by way of the management server, the management server will similarly re-encrypt the trapdoors and publish them to the cloud server. Upon receiving the trapdoor T , the cloud server searches the encrypted index I of every information owner and returns the corresponding set of encrypted documents. to enhance the file retrieval accuracy and save conversation cost, a statistics person might tell the cloud server a parameter k and cloud server might return the top-k relevant documents to the information user. as soon as the records user gets the top-ok encrypted files from the cloud server, he's going to decrypt those returned documents. observe that how to gain decryption talents are out of the scope of this paper; some extremely good work concerning this hassle can be located.

Chance model:

In our risk version, we count on the management server is trusted. the administrative server can be any trusted 0.33 party, e.g., the certificates Authority

inside the Public Key Infrastructure, the aggregation and distribution layer, and the 1/3 birthday celebration auditor. Facts owners and data customers who handed the authentication of the management server also are trusted. but, the cloud server isn't depended on. Instead, we deal with the cloud server as 'curious however sincere' which is the same as in preceding works. The cloud server follows our proposed protocol, but it's far keen to obtain the contents of encrypted files, keywords, and relevance rankings. Observe that preserving the get entry to sample, i.e., the list of returned files, is extremely expensive since the set of rules has to 'contact' the entire record set. We do no longer purpose to defend it in this work for performance issues.

Design goals and security Definitions:

To enable privacy retaining ranked multi-keyword search within the multi-owner and multi-person cloud environment, our machine layout have to concurrently satisfy safety and performance desires.

• Ranked Multi-keyword search over Multiowner: The proposed scheme should allow multi-key-word seek over encrypted documents which would be encrypted with different keys for different data proprietors. It also desires to allow the cloud server to rank the search effects among exclusive information owners and return the top-okay outcomes.

• Information proprietor scalability: The proposed scheme ought to permit new facts proprietors to go into this gadget without affecting different facts owners or data customers, i.e., the scheme have to support statistics proprietor scalability in a plug-and-play model.

• Records user revocation: The proposed scheme should ensure that simplest authenticated facts users can perform correct searches. Furthermore, once a facts consumer is revoked, he can no longer carry out accurate searches over the encrypted cloud records.

• Security desires: The proposed scheme must attain the subsequent safety desires: 1) keyword Semantic safety (Definition 1). we will prove that PRMSM achieves semantic security in opposition to the chosen keyword attack. 2) key-word secrecy (Definition 2). for the reason that adversary A can understand whether an encrypted keyword suits a trapdoor, we use the weaker security purpose (i.e., secrecy), that is, we have to make sure that the chance for the adversary A to infer the actual fee of a

2.4 Notations

- $\mathcal{O}$: the data owner collection, denoted as a set of m data owners $\mathcal{O} = (O_1, O_2, \ldots, O_m)$.
- $\mathcal{F}_i$: the plaintext file collection of O_i, denoted as a set of n data file $\mathcal{F}_i$=($F_{i,1}$, $F_{i,2}$,…,$F_{i,n}$).
- $\mathcal{C}_i$: the ciphertext file collection of $\mathcal{F}_i$, denoted as $\mathcal{C}_i$=($C_{i,1}$, $C_{i,2}$,…,$C_{i,n}$).
- $\mathcal{W}$: the keyword collection, denoted as a set of u keywords $\mathcal{W} = (w_1, w_2, \ldots, w_u)$.
- $\widehat{\mathcal{W}}_i$: O_i's encrypted keyword collection of $\mathcal{W}$, denoted as $\widehat{\mathcal{W}}_i = (\widehat{w}_{i,1}, \widehat{w}_{i,2}, \ldots, \widehat{w}_{i,u})$.
- $\widetilde{\mathcal{W}}$: the subset of $\mathcal{W}$ which represents queried keywords, denoted as $\widetilde{\mathcal{W}} = (w_1, w_2, \ldots, w_q)$.
- $\mathcal{T}_{\widetilde{\mathcal{W}}}$: the trapdoor for $\widetilde{\mathcal{W}}$, denoted as $\mathcal{T}_{\widetilde{\mathcal{W}}} = (T_{w_1}, T_{w_2}, \ldots, T_{w_q})$.
- $\mathcal{S}_{i,j,t}$: the relevance score of tth keyword to jth file of ith data owner.

keyword is negligibly greater than randomly guessing. 3) Relevance score secrecy. We should make sure that the cloud server cannot infer the actual fee of the encoded relevance rankings.

Definition 1: Given a probabilistic polynomial time adversary *A*, he asks the challenger *B* for the cipher-text of his submitted keywords for polynomial times. Then *A* sends two keywords *w*0 and *w*1, which are not challenged before, to *B*. *B* randomly sets $\mu \in \{0, 1\}$, and returns an encrypted keyword ˆ *w*_ to *A*. *A* continues to ask *B* for the cipher-text of keyword *w*, the only restriction is that *w* is not *w*0 or *w*1. Finally, *A* outputs its guess μ' for μ. We define the advantage that A breaks *PRMSM* as AdvA=Pr[$\mu = \mu'$] – 1/. If AdvA is negligible, we say that PRMSM is semantically secure against the chosen-keyword attack.

Definition 2: Given a probabilistic polynomial time adversary A, he asks the challenger B for the cipher-text of his queried keywords for t times. Then B randomly chooses a keyword w*, encrypts it to ˆ w*, and sends ˆ w* to A. A outputs its guess w′ for w*, and wins if w′ = w*. We define the probability that A breaks keyword secrecy as AdvA=Pr [w′ = w*].

We say that PRMSM achieves keyword secrecy if $Adv_A = 1$ u−t + ε, where ϵ is a negligible parameter, t denotes the number of keywords that has known, and u denotes the size of keyword dictionary.

3 PRELIMINARIES:

Before we introduce our detailed construction, we first briefly introduce some techniques that will be used in this paper.

Bilinear Map:

Let G and G1 denote two cyclic groups with a prime order p. We further denote g and g1 as the generator of G and G1, respectively.

4. Information person authentication:

To prevent attackers from pretending to be legal records users appearing searches and launching statistical attacks based at the search result, data customers need to be authenticated before the management server reencrypts trapdoors for records customers. conventional authentication techniques regularly comply with three steps. first, records requester and facts authenticator share a mystery key, say, k0. 2d, the requester encrypts his in my opinion identifiable data d0 using k0 and sends the encrypted information (d0)k0 to the authenticator. third, the authenticator decrypts the obtained information with k0 and authenticates the decrypted statistics. but, this technique has predominant drawbacks. first, since the mystery key shared between the requester and the authenticator stays unchanged, it is straightforward to incur replay attack. 2nd, once the secret key is discovered to attackers, the authenticator can not distinguish between the criminal requester and the attackers; the attackers can pretend to be prison requesters with out being detected. On this segment, we first supply an overview of the records user authentication protocol. then, we introduce how to obtain secure and green statistics consumer

authentication. ultimately, we exhibit the way to come across illegal searches and the way to allow cozy and green records consumer revocation.

Review:

Now we deliver an example to demonstrate the primary concept of the consumer authentication protocol (the certain protocol is elaborated within the following subsections). Count on alice desires to be authenticated by using the management server, so she starts a verbal exchange with the server. The server then authenticates the contents of the communication. if the contents are authenticated, both alice and the server will generate the initial secret key consistent with the conversation contents. after the initialization, to be authenticated efficiently, alice has to offer the historic data in their conversations. if the authentication is a hit, both alice and the administration server will alternate their secret keys in accordance the contents of the conversation. in this manner, the secret keys maintain changing dynamically; without understanding the suitable historical facts, an attacker cannot begin a successful verbal exchange with the management server.

User authentication: Earlier than we introduce the dynamic key technology method and the authentication protocol, we first introduce the format of the authentication statistics. as proven in fig. 2, the authentication statistics consists of 5 elements. The request counter subject records the quantity of search requests that the facts consumer has submitted. The last request time subject asks the information user to provide the historical facts of his previous request time. The personally identifiable information (e.g., passport quantity, telephone wide variety) field is used to perceive a specific records user, at the same time as the random quantity and crc subject are similarly used to test whether or not the authentication records has been tampered with the key point of a successful authentication is to provide both the dynamically changing secret keys and the historical data of the corresponding data user. Let $k_{i;j}$ denotes the secret key shared between administration server and the jth data user U_j after i instances of search requests, and $d_{i;j}$ denotes the authentication data for the $(i+1)$th request of U_j. Our authentication protocol runs in the following six steps.

A. Data user Uj prepares his authentication data di;j, i.e., U_j needs to fill in all the fields of authentication data based on his historical data.

B. Data user U_j encrypts $d_{i;j}$ with the current secret key ki;j and submits the encrypted authentication data (di;j)ki;j to the administration server.

C. After submitting the authentication data, the data user Uj generates another secret key ki+1;j = ki;j $\oplus$ H(di;j), and stores both ki;j and ki+1;j .

D. Upon receiving U_j 's encrypted authentication data, the administration server decrypts it with $k_{i;j}$.

E. The administration server checks the request counter, last request time, personally identifiable data and CRC, respectively. If the authentication succeeds, the administration server first generates a new secret key ki+1;j = ki;j $\oplus$H(di;j), then he replies a confirmation data $d_{i+1;j}$, and encrypts it with ki+1;j . Otherwise, the administration server encrypts $d_{i+1;j}$ with secret key ki;j .

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 11
July 2016

F. Upon receiving a reply from the administration server, the data user *Uj* will try to decrypt it with ki+1;j . If the decrypted data contains the confirmation data, the authentication is successful. Otherwise, the authentication is regarded as being unsuccessful. The data user deletes the new generated secret key *ki*+1*;j* and considers whether to start another authentication.

Illegal seek Detection:

In our scheme, the authentication technique is protected through the dynamic mystery key and the ancient facts. We assume that an attacker has efficaciously eavesdropped the name of the game key k0;j of Uj . Then he has to assemble the authentication statistics; if the attacker has now not efficiently eavesdropped the historical records, e.g., the request counter, the ultimate request time, he can't construct the best authentication statistics. therefore this unlawful action will soon be detected by means of the administration server. similarly, if the attacker has efficaciously eavesdropped all data of Uj , the attacker can successfully construct the authentication information and fake himself to be Uj without being detected through the management server. However, once the felony information consumer Uj plays his seek, for the reason that mystery key on the management server aspect has modified, there will be contradictory mystery keys among the administration server and the legal data consumer. Consequently, the information user and administration server will quickly locate this unlawful movement.

Request Counter	Last Request Time	Personally Identifiable Data	Random Number	CRC

Fig. 2: Format of Authentication Data

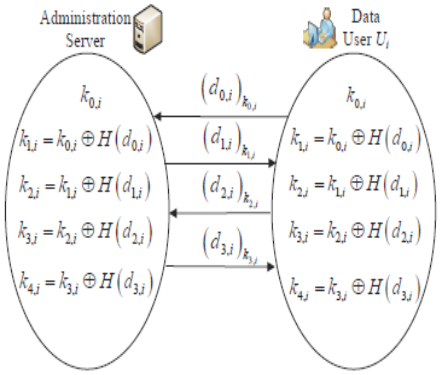


Fig. 3: Example of data user authentication and dynamic secret key generation

Information person Revocation:

Extraordinary from previous works, information person revocation in our scheme does not want to re-encrypt and update big quantities of information saved on the cloud server. Rather, the adminstration server simplest desires to update the secret facts Sa stored at the cloud server. As will be targeted in the subsequent segment, Sa = gka1·ka2·ra , in which ka1 and ka2 are the name of the game keys of the management server, and ra is randomly generated for each update operation. consequently, the preceding trapdoors will be expired. Additionally, with out the help of the management server, the revoked information person cannot generate the correct trapdoor Twh. Therefore, a records user cannot carry out correct searches as soon as he is revoked.

MATCHING DIFFERENT KEY ENCRYPTED KEYWORDS: Numerous data proprietors are regularly concerned in sensible cloud applications. For privateness concerns, they

would be reluctant to proportion mystery keys with others. Rather, they prefer to use their personal mystery keys to encrypt their sensitive records (keywords, documents). When key phrases of various information owners are encrypted with distinct secret keys, the approaching question is a way to discover exceptional-key encrypted key phrases among more than one facts proprietors. on this section, to allow secure, green and handy searches over encrypted cloud statistics owned with the aid of multiple records owners, we systematically design schemes to achieve the subsequent three necessities:

First, distinct information proprietors use unique secret keys to encrypt their keywords. 2d, authenticated statistics customers can generate their trapdoors without knowing those secret keys. 0.33, upon receiving trapdoors, the cloud server can discover the corresponding keywords from one of a kind statistics owners' encrypted keywords without understanding the actual price of keywords or trapdoors.

Evaluate: Now we gift an example to demonstrate the main concept of our key phrases matching protocol (the distinctive protocol is elaborated inside the following subsections). Assume Alice wants to use the cloud to shop her file F, she first encrypts her report F, and receives the cipher-text C.

Keyword Encryption:

For keyword encryption, the following conditions should be satisfied: first, different data owners use their own secret keys to encrypt keywords. Second, for the same keyword, it would be encrypted to different cipher-texts each time. These properties benefit our scheme for two reasons.

$$\hat{w}_{i,h} = \left(g^{k_{i,w}\cdot r_o \cdot H(w_{i,h})}, g^{k_{i,w}\cdot r_o}\right) \quad (1)$$

$$E_a = \left(E_{a'} \cdot g^{k_{a1}}\right)^{k_{a2}} \quad (2)$$

$$\begin{aligned} & \hat{e}\,(S_a, T_2) \\ = \; & \hat{e}\left(g^{r_a \cdot k_{a1} \cdot k_{a2}}, g^{r_u \cdot k_{a1}}\right) \quad (5) \\ = \; & \hat{e}(g,g)^{r_a \cdot k_{a1} \cdot k_{a2} \cdot r_u \cdot k_{a1}} \end{aligned}$$

Trapdoor Generation: To make the statistics users generate trapdoors securely, without difficulty and efficaciously, our proposed scheme need to satisfy important conditions. first, the data consumer does now not want to ask a massive amount of records proprietors for secret keys to generate trapdoors. Second, for the same keyword, the trapdoor generated each time should be distinctive.

$$T'_{w_{h'}} = \left(g^{H(w_{h'})\cdot r_u}, g^{r_u}\right) \quad (3)$$

$$\begin{aligned} & \hat{e}\,(E_a, T_3) \\ = \; & \hat{e}\left(\left(g^{k_{i,w}\cdot r_o \cdot H(w_{i,h})} \cdot g^{k_{a1}}\right)^{k_{a2}}, g^{r_u \cdot k_{a1} \cdot r_a}\right) \\ = \; & \hat{e}(g,g)^{(k_{i,w}\cdot r_o \cdot H(w_{i,h}) + k_{a1}) \cdot k_{a2} \cdot r_u \cdot k_{a1} \cdot r_a} \quad (6) \\ = \; & \hat{e}(g,g)^{k_{i,w}\cdot r_o \cdot H(w_{i,h}) \cdot k_{a2} \cdot r_u \cdot k_{a1} \cdot r_a} \cdot \hat{e}\,(S_a, T_2) \\ = \; & \hat{e}\left(g^{k_{i,w}\cdot r_o}, g^{H(w_{i,h}) \cdot k_{a2} \cdot r_u \cdot k_{a1} \cdot r_a}\right) \cdot \hat{e}\,(S_a, T_2) \\ = \; & \hat{e}\,(E_o, T_1) \cdot \hat{e}\,(S_a, T_2) \end{aligned}$$

$$T_{w_{h'}} = \left(g^{H(w_{h'})\cdot r_u \cdot k_{a1} \cdot k_{a2} \cdot r_a}, g^{r_u \cdot k_{a1}}, g^{r_u \cdot k_{a1} \cdot r_a}\right) \quad (4)$$

Keywords Matching among Different Data Owners:

The cloud server stores all encrypted files and keywords of different data owners. The administration server will also store a secret data like that of S_a = *gka*1 · *ka*2 · *ra* on the cloud server. Upon receiving a query request, the cloud will search over the data of all these data owners. The cloud processes the search request in two steps. First, the cloud matches the queried keywords from all keywords stored on it, and it gets a candidate file set. Second, the cloud ranks files in the candidate file set and finds the most top-*k* relevant files. We introduce the matching strategy here, while leaving the task of introducing the ranking strategy in the next section. When the cloud obtains the trapdoor T_{wh} and encrypted keywords (Eo, Ea), he first computes

Then he can judge whether *wh*′ = *wi*;*h* (i.e., an encrypted keyword is located) holds if the following equation is true.

6. PRIVATENESS RETAINING RANKED SEEKS:

The aforementioned phase allows the cloud fit the queried keywords, and acquires a candidate file set. but, we can not genuinely go back un-differential documents to statistics users for the following two motives. First, returning all candidate documents might purpose abundant verbal exchange overhead for the whole system. 2nd, records users would simplest concern the top-ok applicable documents similar to their queries. on this phase, we first elucidate an order and privacy retaining encoding scheme. Then we illustrate an additive order preserving and privateness maintaining encoding scheme. sooner or later, we practice the proposed scheme to encode the relevance scores and gain the pinnacle-ok search results.

x	1	2	3	4	5
f(*x*)	100-1000	1100-1800	2000-4200	4300-5000	5100-7000

Fig. 4: An example of Order Preserving and Privacy Preserving Function

ORDER AND PRIVACY

$$F^{y}_{aoppf}(x) = \sum_{0 \le j,k \le \tau} A_{j,k} \cdot m(x,j) \cdot m(y,k) + r_{aof} \quad (8)$$

RETAINING FEATURE:

To rank the relevance score at the same time as preserving its privateness, the proposed feature should fulfill the subsequent situations. 1) This

$$\mathcal{V}_{i,j,t} = F^{H_i(i)}_{aoppf}(\mathcal{S}_{i,j,t}) \quad (9)$$

function need to hold the order of records, as this helps the cloud server determine which report is extra

$$F^{y}_{oppf}(x) = \sum_{0 \le j,k \le \tau} A_{j,k} \cdot m(x,j) \cdot m(y,k) + r_f \quad (7)$$

applicable to a sure key-word, consistent with the encoded relevance rankings. 2) This feature ought to now not be discovered by way of the cloud server in order that cloud server could make comparisons on encoded relevance scores without understanding their actual values. 3) specific statistics owners have to have exceptional functions such that revealing the encoded fee of a records proprietor could not cause the leakage of encoded values of other facts owners. to be able to satisfy circumstance 1,we introduce a records processing part m(x, ·), which preserves the order of x. to meet situation 2, we introduce a worrying element rf which facilitates save you the cloud server from revealing this feature. to satisfy situation three, we use m(x, ·) to system the id of records proprietors. So this characteristic belongs to the subsequent characteristic family:

Additive Order and privacy retaining feature to correctly perform the cozy ranked multi keyword search, the sum of any two encoded relevance scores must nevertheless be ordered and privateness preserved (we use the sum of encoded relevance score to assess the relevance between a report and multiple key phrases in this paper). to satisfy this condition, we in addition design an additive order and privateness maintaining characteristic family based on Eq. 7:

Encoding relevance rankings: With the properly-designed residences of F_{aoppf} the cloud server could make a evaluation amongst encoded relevance ratings for the identical information owner. But, seeing that unique records proprietors encode their relevance ratings with extraordinary capabilities in F_{aoppf}, the cloud server can't make a comparison between encoded relevance ratings for different statistics owners. To clear up this problem, we outline:

$$\mathcal{T}_{i,j,t}(y) = F^{y}_{aoppf}(\mathcal{S}_{i,j,t}) \quad (10)$$

$$\mathcal{V}_{i,1} = \mathcal{V}_{i,1,m} + \mathcal{V}_{i,1,n} \quad (11)$$

$$\mathcal{V}_{i,2} = \mathcal{V}_{i,2,m} + \mathcal{V}_{i,2,n} \quad (12)$$

$$\mathcal{V}_{i,1} = \mathcal{V}_{i,1,m} + \mathcal{V}_{i,1,n} \quad (13)$$

$$\mathcal{T}_{j,2}(y) = \mathcal{T}_{j,2,m}(y) + \mathcal{T}_{j,2,n}(y) \quad (14)$$

Rating search outcomes: On this paper, we use the sum of the relevance rankings because the metric to rank seek outcomes. Now, we introduce the strategies of ranking seek effects based totally on the encoded relevance rankings.

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 11
July 2016

$$T_{w_{h'}} = \left(g^{H(w_{h'})\cdot r_u \cdot k_{a1}\cdot k_{a2}\cdot r_a}, g^{r_u\cdot k_{a1}}, g^{r_u\cdot k_{a1}\cdot r_a}\right) \quad (15)$$

7. SAFETY ANALYSIS:

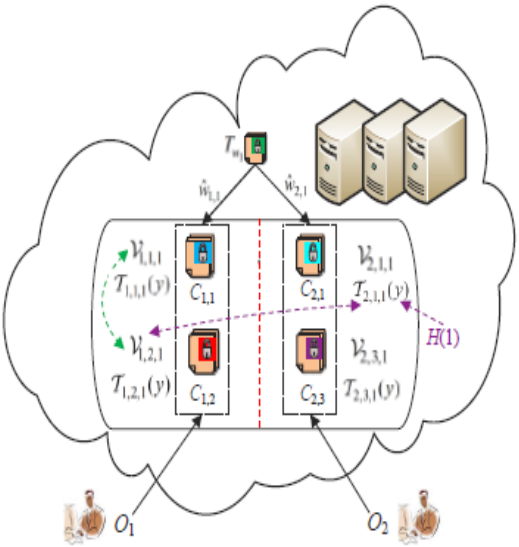


Fig. 5: Example of ranking search results

In this section, we provide step-by means of-step security analyses to demonstrate that the safety necessities had been glad for the data files, the key phrases, the queries, and the relevance scores.

Records files: The records files are protected by means of symmetric encryption earlier than upload. so long as the encryption algorithm is no longer breakable, the cloud server cannot recognize the information.

Key phrases: We formulate the safety goals accomplished by means of PRMSM with the subsequent two theorems.

$$F_{aopp}^{(H_i(i))}(s) = \sum_{0\le j,k\le \tau} A_{j,k}\cdot m(s,j)\cdot m(H_i(i),k) + r_{aof} \quad (16)$$

Theorem 2: Given the DDH (Decisional DiffieHellman) assumption: PRMSM is semantically relaxed in opposition to the selected keyword assault underneath the selective safety version.

Evidence: See Appendix B.

Theorem 3: Given the DL (Discrete Logarithm) assumption, PRMSM achieves keyword secrecy inside the random oracle version.

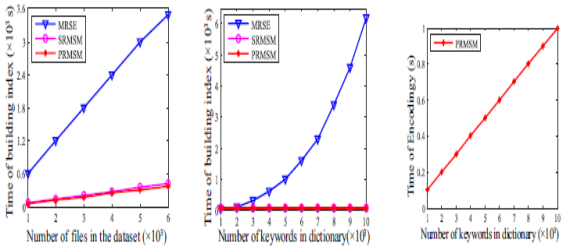
Evidence: See Appendix C.

TRAPDOORS: Recall the trapdoor production formulation

$$T_{w_{h'}} = \left(g^{H(w_{h'})\cdot r_u \cdot k_{a1}\cdot k_{a2}\cdot r_a}, g^{r_u\cdot k_{a1}}, g^{r_u\cdot k_{a1}\cdot r_a}\right) \quad (15)$$

Relevance ratings: In our scheme, relevance ratings are encoded with Additive Order and privateness keeping features. Now we examine the safety of additive order and privateness retaining capabilities. eight performance assessment On this segment, we measure the performance of PRMSM, and compare it with its previous version, comfortable Ranked Multi-keyword search for more than one statistics proprietors in cloud computing (SRMSM), and the stateof- the-art, privacy-maintaining Multi-keyword Ranked search over Encrypted cloud records (MRSE), facet by means of aspect. on account that MRSE is most effective suitable for the single proprietor model, our PRMSM and SRMSM not most effective work nicely in multi-proprietor settings, but additionally outperform MRSE on many aspects.

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 11
July 2016



(a) For different size datasets with the same dictionary, u = 4000.

(b) For the same dataset with different dictionary sizes, n = 1000.

(c) Time cost of encoding relevance score (a subroutine of building index)

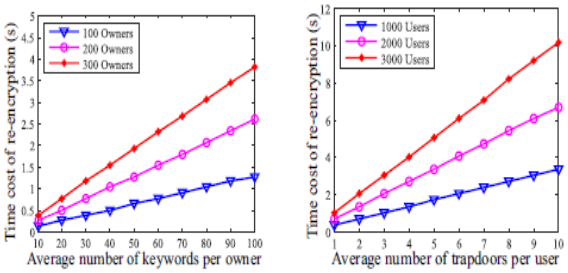
Fig. 6: Time cost of index construction.

Evaluation Settings: We conduct overall performance experiments on actual records set, the internet Request For feedback dataset (RFC). We use airtight word Frequency Counter to extract key phrases from every RFC record. After the keyword extraction, we compute keyword records together with the key-word frequency in each report, the length of each document, the variety of documents containing a selected keyword, and many others. We in addition calculate the relevance score of a keyword to a document based on these facts. The file length and key-word frequency of this records set can be visible.

Evaluation Consequences:

Index construction:

Fig. 6(a) shows that, given the identical key-word dictionary (u=4000), time of index construction for these schemes increases linearly with an growing wide variety of documents, while SRMSM and PRMSM spend a whole lot much less time on index construction. Fig. 6(b) demonstrates that, given the identical quantity of files (n=1000), SRMSM and PRMSM eat a great deal much less time than MRSE on constructing indexes. Moreover, SRMSM and PRMSM are insensitive to the scale of the keyword dictionary for index creation, whilst MRSE suffers a quadratic growth with the size of key-word dictionary increases. Fig. 6(c) indicates the encoding efficiency of our proposed AOPPF. The time spent on encoding increases from zero.1s to 1s while the quantity of key phrases increases from one thousand to 10000. This time price can be ideal.

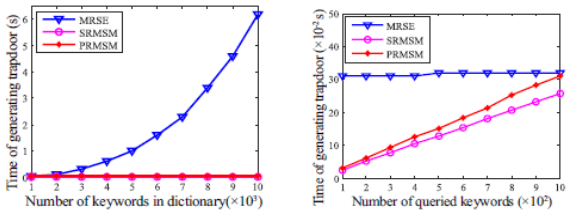


(a) Time cost of re-encrypting keywords

(b) Time cost of re-encrypting trapdoors.

Fig. 8: Time cost of the administration server.

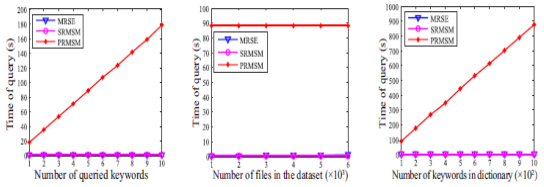
Trapdoor generation: Compared with index construction, trapdoor era consumes noticeably much less time. Fig. 7(a) demonstrates that, given the equal variety of queried key phrases (q=one hundred), SRMSM and PRMSM are insensitive to the scale of



(a) For different size of key-word dictionary with the same number of queried keywords, q = 100.

(b) For different number of queried keywords with the same size of keyword dictionary, u = 4000.

Fig. 7: Time cost of generating trapdoors.



(a) For different size of queried keywords with the same size of data set, n=2000 and key-word dictionary, u = 100.

(b) For different size of data set with the same size of queried keywords, q=5 and keyword dictionary, u = 100.

(c) For different size of key-word dictionary with the same size of queried keywords, q=5 and data set, n = 2000.

Fig. 9: Time cost of search.

key-word dictionary on trapdoor era and consumes zero.026s and 0.031s, respectively. in the meantime, MRSE increases from 0.04s to six.2s. Fig. 7(b) suggests that, given the identical number of dictionary size (u=4000), when the range of queried keywords increases from a hundred to a thousand, the trapdoor generation time for MRSE is 0.31s, and stays unchanged. While SRMSM will increase from zero.024s to zero.25s, PRMSM increases from zero.031s to 0.31s. We study that PRMSM spends a little extra time than SRMSM on trapdoor generation; the reason is that PRMSM introduces an extra variable to make certain the randomness of trapdoors.

Re-encryption by way of the management server: Fig. 8(a) illustrates the re-encryption time fee of the management server in PRMSM. As we are able to see, for the same average quantity of keywords in step with owner, the more data proprietors are involved, the greater time is spent on re-encryption. While there are three hundred data proprietors, each data proprietor has a hundred key phrases; we need 3.8s to re-encrypt those keywords, which is acceptable. Fig. 8(b) demonstrates the time price of re-encrypting trapdoors. We examine that, for the same average range of trapdoors according to consumer, the more data users that publish trapdoors, the greater time could be spent on re-encryption. whilst there are a thousand information customers who simultaneously put up records, every records user has 10 trapdoors; we simplest want 3.34s for re-encryption.

SEEK:

From Fig. 9, we study that, PRMSM spends more time for looking. The fundamental purpose is that, the pairing operation used in PRMSM wishes more time. As we are able to see from Fig. nine(a) and Fig. 9(c), the more keywords current in the cloud server, the more time is required for pairing operation. Fig. 9(b) confirms that after the variety of key phrases stored on the cloud server stays a constant, PRMSM will not boom even though the number of documents increases. Though PRMSM spends highly more time, this commentary additionally confirms that the searching operation should be outsourced to the cloud server.

9. Associated Paintings: On this section, we evaluation three classes of labor: searchable encryption, relaxed key-word seek in cloud computing, and order maintaining encryption.

Searchable Encryption: The earliest try of searchable encryption became made by using track et al. They endorse to encrypt every word in a record independently and permit the server to locate whether a single queried key-word is contained within the file without knowing the exact phrase. This thought is extra of theoretic pursuits because of high computational prices. Goh et al. suggest building a keyword index for each report and using Bloom clear out to boost up the search. Curtmola et al. endorse building indices for each key-word, and use hash tables as an alternative technique to searchable encryption. The primary public key scheme for keyword search over encrypted facts is offered.

Further enrich the quest functionalities of searchable encryption by way of offering schemes for conjunctive key-word seek. The searchable encryption cares more often than not approximately unmarried key-word search or boolean keyword search. Extending these techniques for ranked multi-keyword search will incur heavy computation and garage charges.

Secure Keyword Search In Cloud Computing: The privacy worries in cloud computing inspire the study on secure key-word search. Wang et al. first defined and solved the at ease ranked keyword search over encrypted cloud statistics. They proposed a scheme that returns the top-okay applicable documents upon a unmarried key-word seek. Cao et al. and solar et al. prolonged the relaxed keyword search for multi-keyword queries.

10. CONCLUSIONS:

In this paper, we explore the problem of relaxed multi-keyword search for more than one data owners and a couple of records customers within the cloud computing environment. Exceptional from earlier works, our schemes allow authenticated data customers to reap comfortable, convenient, and green searches over a couple of facts owners' facts. To correctly authenticate facts customers and locate attackers who steal the name of the game key and carry out illegal searches, we suggest a singular dynamic secret key technology protocol and a new statistics user authentication protocol. To permit the cloud server to carry out relaxed seek amongst a couple of owners' records encrypted with exclusive secret keys, we systematically assemble a novel cozy search protocol. To rank the search outcomes and maintain the privateness of relevance ratings among keywords and documents, we advocate a novel Additive Order and privateness retaining characteristic family. Moreover, we display that our method is computationally efficient, even for big records and keyword units. As our future work, on one hand, we will don't forget the problem of secure fuzzy key-word search in a multi-owner paradigm. however, we plan to implement our scheme on the economic clouds.

REFERENCES:

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Communication of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[2] C. Wang, S. S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacypreserving public auditing for secure cloud storage," *Computers, IEEE Transactions on*, vol. 62, no. 2, pp. 362–375, 2013.

[3] D.Song, D.Wagner, and A.Perrig, "Practical techniques for searches on encrypted data," in *Proc. IEEE International Symposium on Security and Privacy (S&P'00)*, Nagoya, Japan, Jan. 2000, pp. 44–55.

[4] E. Goh. (2003) Secure indexes. [Online]. Available: http://eprint.iacr.org/

[5] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," in *Proc. ACM CCS'06*, VA, USA, Oct. 2006, pp. 79–88.

[6] D. B. et al., "Public key encryption with keyword search secure against keyword guessing attacks without random oracle," *EUROCRYPT*, vol. 43, pp. 506–522, 2004.

[7] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," in *Proc. Applied Cryptography and Network Security (ACNS'04)*, Yellow Mountain, China, Jun. 2004, pp. 31–45.

[8] L. Ballard, S. Kamara, and F. Monrose, "Achieving efficient conjunctive keyword searches over encrypted data," in *Proc. Information and Communications Security (ICICS'05)*, Beijing, China, Dec. 2005, pp. 414–426.

[9] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure ranked keyword search over encrypted cloud data," in *Proc. IEEE Distributed Computing Systems (ICDCS'10)*, Genoa, Italy, Jun. 2010, pp. 253–262.

Mr. T.SRIKANTH was born in India in the year of 1984. He received B.Tech degree in the year of 2007 From ANU & M.Tech PG in the year of 2012from JNTUK. He was expert in OBECT ORIENTED ANALYSIS AND DESIGN, DATA MINING, COMPUTER GRAPHICS and LINUX PROGRAMMING subjects. He is currently working as An Asst. Professor in CSE Department in Malla Reddy Institute of Technology, Maisammaguda, Dhulapally Post, Secunderabad and Telengana State, India.

Mail id : srikanth282@gmail.com

Ms. B.MANJULA was born in India. She pursuing M.Tech degree in Computer Science & Engineering in CSE Department in MallaReddy Institute of Technology, Maisammaguda, Dhulapally Post, Secunderabad and Telengana State, India.

Mail id: manjulabangari@gmail.com