Asian Journal of Computer Science and Technology
ISSN: 2249-0701 Vol.8 No.S3, 2019, pp. 122-127

Encrypted Mobile Cloud Data Searching With Efficient Traffic and Energy Saving Method

M. Jamuna[1] and A. Supriya[2]
[1]Student, [2]Assistant Professor, [1&2]Department of Computer Science and Engineering,
School of Engineering and Technology, Sri Padmavati MahilaVisva Vidyalayam, Tirupati, Andhra Pradesh, India
E-Mail: jamunajamu30@gmail.com, supriyaaddanki@gmail.com

***Abstract* - Cloud storage provides a convenient, massive, and scalable storage at low value; however information privacy could be a major concern that prevents users from storing files on the cloud confidingly. a technique of enhancing privacy from information owner purpose of read is to cipher the files before outsourcing them onto the cloud and decode the files when downloading them. However, encryption could be a heavy overhead for the mobile devices, and information retrieval method incurs an advanced communication between the information user and cloud. Commonly with restricted bandwidth capability and restricted battery life, these problems introduce significant overhead to computing and communication as well as the next power consumption for mobile device users that makes the encrypted search over mobile cloud terribly difficult. During this paper, we tend to propose traffic and energy saving encrypted search (TEES), a bandwidth and energy efficient encrypted search design over mobile cloud. The planned design offloads the computation from mobile devices to the cloud, and that we any optimize the communication between the mobile clients and also the cloud. It's demonstrated that the information privacy doesn't degrade once the performance sweetening ways square measure applied. Our experiments show that TEES reduces the computation time by twenty three to forty six p.c and save the energy consumption by 35 to 55 percent per file retrieval; meanwhile the network traffics throughout the file retrievals also are considerably reduced.**
Keywords: **Mobile Cloud Storage, Searchable Data Encryption, Energy Efficiency, Traffic Efficiency**

I. INTRODUCTION

CLOUD storage system may be a service model during which information are maintained, managed and backed remotely on the cloud aspect, and in the meantime information keeps offered to the users over a network[1],[2].Mobile Cloud Storage (MCS), denotes a family of progressively well-liked on-line services, and even acts because the primary files storage for the mobile devices. To this finish, we have a tendency to introduce Traffic and Energy saving Encrypted Search (TEES) design for mobile cloud storage applications. Traffic quantity for retrieving information from encrypted cloud storage [2],[3].Besides the energy and traffic efficiencies, MCS, by adding noise in Term Frequency (TF) distribution operate and keeping the Order protective encoding (OPE) attributes.

TEES employs the design over ancient encrypted search procedure, and our comprehensive experiments prove the TEES has following advantages as compared with the normal complicated encrypted search procedure:

1. TEES reduces the energy consumption by 35 fifty five p.c by offloading the computation of the relevance scores to the cloud server[4].This reduces the computing employment on the mobile device facet while at constant time considerably rushing up the mobile file access speed.
2. With a simplified search and retrieval method, TEES reduces the network traffic for the communication of the selected index, and reduces the file retrieval time by twenty three forty six p.c in our experiments.
3. In implementing TEES redistributes the encrypted index to avoid statistics data leak, and wraps keywords adding noise so as to render them indistinguishable to the attackers. Security analysis shows that the protection level of TEES is secured and enhanced for MCS wireless communication channels.

II. FILE RETRIEVAL IN CLOUD STORAGE

A. Traditional Encrypted Search over Cloud Data

Traditional cloud storage system design and general procedures that include: file/index encryption by the information owner, outsourcing the information to the cloud storage, and encrypted information search/retrieval procedure of the information users in cloud computing.

1. File/Index cryptography

The data owner initial executes the preprocessing and categorization work. He ought to invert files that are selected to store on the cloud, for text search engines [5]. Every word in these files undergoes stemming to retain the word stem. After this step, the data owner encrypts and hashes every term (word stem) to fix its entry in the index. The index is then created by the data owner. Finally, the data owner encrypts the index and stores it into the cloud server, together with the encrypted file set.

2. Data Search and Retrieval after Authentication

A data user will solely access a file once being genuine by the data owner. Within the method of authentication, the information user sends his identity to the information

owner. The information owner sends the encrypted keys back if the user may be a legal user.

In the method of search and retrieval, the cloud server helps the users to search out the top-k relevant files for a given keyword without decrypting Searches incur following the steps

1. Associate degree genuine user stems the keyword to be queried, encrypts it with the keys and hashes it to induce its entry within the index [5]. Then the encrypted keyword is sent to the cloud server.
2. On receiving the encrypted keyword, the cloud server initial searches for it within the index. Then the index associated with this keyword is distributed back to the data user.
3. The information user calculates the connectedness scores with the selected index to search out the top-k relevant files and sends a follow-up request to the cloud server in order to retrieve the files.
4. The position of those files is chosen and that they square measure sent back to the information user from the cloud server.
5. The information user decrypts the files and recovers the original information.

We have a tendency to decision this file retrieval scheme abbreviated as two trip Search (TRS). This scheme provides privacy protection through a sophisticated file retrieval method compared to a straightforward Plain Text Search scheme (PTS)

B. MCS Challenges and Design Principle

1. Effiency Challenges: First, it's obvious that in ancient schemes, the mobile client contains a serious employment for decrypting the chosen index, scheming and ranking the connection scores. [3][4]Second, the two round-trips for every file search and retrieval request, is a heavy burden for mobile devices with restricted information measure and traffic fees

2. Security Challenges: Per the potency challenges in cloud storage mentioned before, we must always then address the security challenges introduced by offloading a part of the calculation onto the cloud. Therefore, the index ought to be keep within the cloud, leading to potential threats for MCS within the following cases:

1. *Statistics Info Leak*: Attackers may get the terms by analyzing the TF table, since Associate in Nursing order conserving encryption technique encrypted TF table produce a high-pitched bar graph of TF values. In other words, term frequency ought to be equally distributed to avoid datum info leak
2. *Keywords-Files Association Leak*: associate degree assaulter may determine question terms by observant queries and results through a wireless channel: because the results of the retrieval is keyword specific, attackers might guess the queried keyword by solely observant the keyword and the results of the retrieval.
3. *Server Info Acquisition*: The cloud server perhaps honest-but-curious and my attempt to learn the underlying plaintext of user's information. [6], [7],. The cloud server can infer and analyze the encrypted index and find additional info, and that we have to be compelled to minimize the information acquisition of the curious cloud server.

Design principle. Our style goal is to realize associate degree economical encrypted search design, whereas each considering these security threats within the changed implementation. Our scheme will offload most of the procedure load to the cloud server.

C. Related Work

1. Encrypted Search Schemes

Over the past recent years, encrypted search has evolved toward the power information sharing with protection of users' privacies. Song et al. raised the question a way to do keyword searches on encrypted information with efficiency. In data Retrieval, term frequency-inverse document frequency (TF-IDF) may be a datum that reflects how vital a word is to a document in an exceedingly assortment or corpus.

Up to now, encrypted search includes Boolean keyword search and stratified keyword search. In Boolean keyword search the server sends back files solely supported the existence or absence of the keywords, while not wanting at their relevancy. Ranked keyword search. Yangtze and Mitzenmacher provided a theme of keyword search, however it doesn't send back the foremost relevant files. In previous work, projected a matched mapping OPE which can lead to *statistics data leak control*. Wang et al. proposed a one-to-many mapping OPE; They enforced a complicate rule for security protection.

2. Power and Traffic Efficiency Improvements Schemes

The previous schemes cannot directly apply to mobile cloud, for achieving economical energy consumption to address the necessary issue for mobile cloud. In recent years several OPE or absolutely homomorphic encryption ways are planned. They proved themselves secure and correct enough for looking encrypted information purpose.

III. TEES SYSTEM DESIGN

To effectively support Associate in Nursing encrypted search theme with a high security level over cloud information, we tend to introduce a replacement architecture that we tend to name TEES [5],[6]. In step with the threats introduced pair of, our aim is to style a sensible resolution for secure encrypted search over mobile cloud storage.

A. The Basic Idea of TEES

The basic plan behind TEES is to dump the calculation and the ranking load of the connection scores to the cloud. There square measure ordinarily 3 main processes:

1. The method of authentication is employed by the info owner to attest the info users.
2. The file set and its index square measure hold on within the cloud when being encrypted by the info owner throughout the preprocessing and categorization stages.
3. The data user searches the files reminiscent of a keyword by causation missive of invitation to the cloud server in the search and retrieval processes.

We currently introduce the elaborate style however TEES addresses the ability potency and therefore the security challenges in modifying these processes.

B. Modified Process of Search and Retrieval

During the preprocessing and categorization stages, the data owner gets a TF table as index and uses order protective encryption to code it. As a result, the cloud server is ready to calculate the relevancy scores and rank them while not decrypting the index. Thus, the changed search and retrieval processes of TEES follow the steps:

1. If a knowledge user desires to retrieve the top-k relevant files based on a keyword, he initial obtains authentication from the data owner then receives the keys to code the keyword.
2. The info user stems the keyword to be queried and encrypts it exploitation the keys.
3. The info user wraps the encrypted keyword into a tuple, adding some noise to avoid datum data leak; this tuple is employed to perform the retrieval.
4. On receiving the wrapped keyword, the cloud server first makes certain that it's accessed by a legal user. If the server unwraps the tuple to recover the entry of the keyword and searches for it within the index.
5. The info user decrypts these files within the mobile shopper and recovers the initial knowledge.

IV. DISCUSSION

A. Performance Efficiency of TEES

The overall design of TEES in which the connectedness scores calculation is offloaded to the cloud, which eases the serious burden on mobile shoppers. The file search and retrieval steps area unit as follows:

1. The info user sends his identity to the info owner and gets the key keys if documented.
2. Associate documented user stems the keyword to be queried, encrypts it with the keys and hashes it to urge its entry in the index.
3. On receiving the encrypted keyword, the cloud server will use the perform of connectedness score calculation to search out the top-k relevant files and sent back to the info user wherever the top-k is designed by the users.
4. The info user decrypts the files and recovers the first data. Thus, the advantages of TEES area unit simply ascertained

B. Reducing the Energy Consumption

Energy may be a precious resource on mobile phones. To define the energy potency h is described wherever Eco denotes the energy consumption per request/response between the user and the server; Eretr stands for the energy consumption throughout file retrieval and action denotes the energy consumption of a mobile device for the relevancy score calculations

$$h_{1/4}=\frac{E_{co}+E_{ret}}{2E_c+E_{retr}+E_w}<100\%$$

Since relevancy score calculation is offloaded to the cloud use to the ORS style of TEES military action is eliminated.ORS compacts the file search and retrieval method into a single spherical (only one Eco). Eretr is identical for each TRS and ORS.

C. Reducing File Search and Retrieval Time

Reducing the execution time of a file search and retrieval transaction is very important for the user expertise. Note that beyond the "one round-trip-time" saved by TEES. Assume that the computing ability of the cloud server and of the mobile purchasers area unit denoted by Ccs and Cm (Cm<< Ccs), severally. Then, the looking work is S, and Tretr denotes the file retrieving time, whereas RTT represents one trip time. The

$$r\ ¼=\frac{RTT + SCcs + Tretr}{2RTT + S\ Cm + Tretr} < 100\%$$

D. Reducing Traffic Overhead

It is important to attenuate the communication overhead so as to ensure that it doesn't cancel the other performance improvements. It's nearly a similar in each TRS and ORS. Let Ckw be the dimensions of the keywords, cathode-ray tube be the network traffic of one communication between the cloud and also the knowledge user, Cf be the overall size of the top-k files, kiss of life be the chosen index transmitted in TRS.

$$z\ ¼ =\frac{Ckw + Crt + Cf}{Ckw +2Crt + Cpr + Cf}<100\%:$$

Overall, the modification of the file search and retrieval processes in TEES style will cause a lot of economical traffic and a small energy consumption. Next section can introduce the elaborated implementation technologies in TEES.

V. TEES IMPLEENTATION FOR SECURITY ENHANCEMENT FOR MOBILE CLOUD

In order to realize security improvement with energy and traffic potency, we tend to implement the modules [6] in TEES exploitation modified routines and new algorithms. Our system is introduced in 3 elements.

A. Redesign of the Data Owner Module

We modify the method of building the index to support the ORS [7] themes by our one-to-many OPE and implement it to control the statistics data leak.

TF-IDF. TF-IDF is that the product of two statistics, term frequency and inverse document frequency (IDF). Within the case of the term frequency tf (t, d), [13] the amount of times that term t happens in document d. If we denote the raw frequency of t by f(t,d), then the simple TF theme is tf(t,d)=f(t,d).

The inverse document frequency may be a [8] live of whether or not the term t is common or rare across all documents. it's obtained by dividing the overall variety of documents by the amount of documents containing the term, so taking the log of that quotient. Then TF-IDF is calculated as

Tf-idf (t,d,D)=tf(t,d)*idf(t,d),

Build Index: In TEES, the info owner starts by collection the files he needs to store into the cloud. Think [8]about a file set F= (F1; F2; . . . ; Fn) containing the amount of | F | files, in which a term set T=(t1; t2; . . . ; tm) and also the variety of | T | terms seem. We tend to produce a table of size | F| * | T | for all thefiles and every one the terms.

Let *GenKey()*be the operate that generates the keys. LetpðÞ be a hash operates that encrypts the terms. In TEES, pðÞis instantiated by a hash operate like MD5. Let "ðÞ be associate coding algorithmic program for the TF values. Once building associate index, it executes following two steps:

1. First, the info owner starts by occupation GenKeyðÞ, to generate a key a to cipher the terms, a key b to encrypt the index, and also the noise > zero;m > 0 to wrap the keywords. He then outputs K ¼ fa; bg andN ¼f;mg.
2. Second, the info owner builds a secure index by occupation Build Index ðK;FÞ (encrypted N is shipped to cloudserver) as represented in algorithmic program

Algorithm 1:Build Index

Input: K, F
Output: I
1: Extract the terms=(t1; t2; . . . ; tm)from the file set F.
2: for ti two T do
3: Get the encrypted term pa(ti) and hash it to urge its entryc(pa(ti)) within the TF table.
4: end for
5: for ti two T and one j jFj do
6: Calculate the term frequency tfij and obtain t~fij ¼ jðS=jFijÞ tfijj.
7: end for
8: cipher "b(t~fij), and store it within the index I.
9: come back I;

In order to extend the entropy of the ciphertext, we adaptively generate the mapping vary consistent with the distribution of the raw TF values and write in code them as represented in Algorithms a pair of and three. Here Ḡ (TF)={G(tf1); . . .;G(tfi)} may be a bound set for all the TF values in a sure TF table, whereas Ḣ (TF) is that the boundary set.

Algorithm 2:Key Generation

Input: TF Table
Output: ~G(TF), ~H (TF)
1: Get the distribution bar graph C of the TF table and acquire TFxas all TF values occur in C.
2: for tfi two TFx do
3: Get the prevalence ci.
4: end for
5: Get C ¼ PjTFxj i¼1 ci.
6: for tfi two TFx do
7: Calculate pi ¼ ci=C.
8: end for
9: for tfi two TFx do
10: if i ¼¼ one then
11: Get G(tfi) =1 and H(tfi) =floor(2B pi).
12: else
13: Get G(tfi)=H(tfi-1)+1 and H(tfi) =H(tfi-1) þ floor(2B pi).
14: end if
15: finish for
16: come back ~G (TF), ~H(TF).

Our rule is order-preserving since a given TF price if1 is encrypted to associate number Eðtf1Þ that is a smaller amount than tfy 1,and tf2 ¼ tf1 þ one is mapped into a replacement vary whose lower boundary is larger than tfy 1. Our one-to-many OPE flattens the TF values bar graph over a file set therefore on make sure the system security,

Algorithm 3: Order Conserving Secret Writing

Input: tf
Output: E(tf)
1: for ti a pair of T and one j jFj do
2: Get E(tfij), E(tfij) R {G(tfij),G(tfij) + 1, . . .;H(tfij)}.
3: end for
4: come back E(tf)
The info owners can even use Advanced Encryption Standard (AES) to cipher the files. *Authentication.* In TEES, the info owner maintains a collection of legal users ("legal set") and a collection of users that may become invalid in when an outlined delay ("overdue set").

B. Redesign of the Data User Module

The data user module is dead on the mobile shoppers' side. The wrap operate of the keywords is enforced to solve the keywords-files association leak. This hash worth is then sent to the cloud server and wont to figure the connexion scores. The wrap operate Wrap () can, first of all produce a random variety r, then build a tuple (h1, h2) supported rule four (> 0;m > 0) Given a keyword w, the info user searches WrapðwÞ ¼ ðh1; h2Þ.

Algorithm 4:Wrap Operate with Noise

Input: w;ʎ, μ
Output: Wrap(w)
1: Stem w and acquire ŵ.
2: Get encrypted term pa(ŵ) associate degreed hash it to urge an entry h ¼
C(pa(ŵ)).
3: produce a random range r.
4: if r < h then
5: Get Wrap(w) = ððh mrÞ2; h2 þ mrÞ.
6: else
7: Wrap(w) =($\mu h2 + \mu\pi$ þ r; ðr mh)2).
8: end if
9: come Wrap(w)

C. Redesign of the Cloud Server Module

We will describe the functions that unwraps and the uncover operate is processed by the server. Upon receiving the tuple Wrap(w)=(h1,h2), the server calls Unwrap((h1,h2)) to urge ¥(ŵ) = h, sarches into the TF table, then sends back the corresponding files. The cloud server sends back the top-k relevant files once ranking the scores victimization this connexion score calculation algorithm,

Algorithm 5: Top-k Ranking Operate

Input: w; k
Output: topFiles
if this request is distributed by a "legal" user then
for each file Fc € F do
Calculate Score (w; Fc).
end for
end if
if this request is distributed by a owed user then
for each file Fc € F do
Calculate Score(w,Fc) however with a warning.
end for
else
Return "No Permission".
end if
Rank the scores to induce prime-k files topFiles ={topF1; topF2; . . . ; topFkg.
return topFiles.
Moreover, if we search encrypted data with multi-keywords, we should sacrifice the search accuracy because most popular OPE does not support multi-keyword well.

VI. SECURITY ANALYSIS AND EVALUATION

In this section, we tend to analyze the safety of TEES supported important security threats mentioned. Concretely, [11] [12] the most necessary principle of the planning is to stop the assaulter from getting any plaintext info regarding our record set or the searched keyword. Our experiments area unit divided into 3 types

A. Statistics Information Leak Control

TEES protects the terms from being determined by analyzing the distribution of the TF [11] values through mobile cloud communication channels. TEES, we have a tendency to encode the TF table with one-to-many order conserving cryptography [7][8][9]. Every TF value is mapped to a mapping vary OPE of TEES reduces the Y-axis vary from thirty-seven K to 250, and enlarge the coordinate axis from about twenty-five to 250.

B. Keywords-Files Association Leak Control

TEES conjointly enhances the safety of the keywords by preventing the assailant from observant the keywords to be searched yet because the results of an enquiry.

C. Server Information Acquisition

We assume that the cloud storage system supplier won't collude with malicious users or intrude users' knowledge on purpose [7][8][9]. The cloud server will infer and analyze the encrypted index and obtain extra info, but it has no intention to switch any necessary knowledge. We tend to use the non-public cloud server from our faculty and assumed it as honest and perform necessary calculations here.

VII. RUNTIME PERFORMANCE EVALUATIONS

In addition to the system security analysis and analysis, we currently judge TEES performance in terms of energy, traffic and file access. We are going to compare its performances to those of TRS and PTS schemes.

A. Experimental Environment

In our experiments, we have a tendency to use an information set of one, 000 files with totally different sizes and a VM within the cloud with twin vCPUs at 2.27 GHz. associate degree mechanical man good phone with a central processing unit at one Gc sends the queries because the mobile shopper of TEES through associate degree about eight M wireless network.

B. Energy Consumption

As energy consumption is crucial for mobile devices, we evaluate TEES energy potency during this section[11].We tend to use Battor a phone power monitor to accurately live

the system energy consumption. Observe that the energy consumption is reduced from 0.08 to 0.036 mAh once looking out and retrieving files of size a hundred K, which implies that ORS saves fifty five p.c energy compared to TRS.

C. File Search and Retrieval Time

We compare the File looking and Retrieval Time (FSRT) for the 3 schemes during this section[10][13].We takes a look at the FSRT for various files with size starting from100 computer memory unit to one MB. The FSRT price of ORS is incredibly near to the one among PTS, implying an awfully low price to security on the mobile device. As an example, TEES saves FSRT by46 % compared to TRS for files of size a hundred computer memory unit, and by 23 % for one MB files.

VIII. CONCLUSION AND FUTUREWORK

In this paper, we have a tendency to develop a brand new design; TEES as an initial arrange to produce a traffic associated energy economical encrypted keyword search tool over mobile cloud storages. We started with the introduction of a basic theme that we have a tendency to compare to previous encrypted search tools for cloud computing and showed their unskillfulness in an exceedingly mobile cloud context. Then we have a tendency to developed associate economical implementation to attain associate encrypted search in an exceedingly mobile cloud. We have planned one keyword search theme to make encrypted knowledge search economical. However, there are still some attainable extensions of our current work remaining. We would prefer to propose a multi-keyword search scheme to perform encrypted knowledge search over mobile cloud in future. As our OPE rule could be a straightforward one, another extension is to search out a strong rule which cannot harm the potency.

REFERENCES

[1] D. Hiemstra, "A probabilistic justification for using tf_idf term weighting in information retrieval", *Int. J. Digital Libraries,* Vol. 3, No. 2, pp. 131–139, 2000.

[2] L. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds: Towards a cloud definition", *ACM SIGCOMM Computer Commun. Rev.*, Vol. 39, No. 1, pp. 50–55, 2008.

[3] D. Huang, "Mobile cloud computing", *IEEE COMSOC Multimedia Commun. Techn. Committee E-Letter,* Vol. 6, No. 10, pp. 27–31, 2011.

[4] C. Wang, N. Cao, K. Ren, and W. Lou, "Enabling secure and efficient ranked keyword search over outsourced cloud data", *IEEE Trans. Parallel Distrib. Syst.*, Vol. 23, No. 8, pp. 1467–1479, Aug. 2012.

[5] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data", *IEEE Trans. Parallel Distrib. Syst.,* Vol. 25, No. 1, pp. 222–233, Jan. 2014

[6] J. Zobel and A. Moffat, "Inverted files for text search engines", *ACM Computer. Surveys,* Vol. 38, No. 2, pp. 6, 2006.

[7] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation", *J. Mach. Learn. Res.,* Vol. 3, pp. 993–1022, 2003.

[8] K. Jones, "Index term weighting", *Inf. Storage Retrieval,* Vol. 9, No. 11, pp. 619–633, 1973.

[9] M. Li, S. Yu, K. Ren, W. Lou, and Y. T. Hou, "Toward privacy assured and searchable cloud data storage services", *IEEE Newt.,* Vol. 27, No. 4, pp. 56–62, Jul./Aug. 2013.

[10] A. Aizawa, "An information-theoretic perspective of tf-idf measures", *Inf. Process. Manage,* Vol. 39, pp. 45–65, 2003.

[11] K. Kumar and Y. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *Computer,* Vol. 43, No. 4, pp. 51–56, 2010.

[12] J. Zobel and A. Moffat, "Exploring the similarity space", *ACM SIGIR Forum,* Vol. 32, No. 1, pp. 18–34, 1998.