www.ijiarec.com

ISSN:2348-2079

Volume-9 Issue-2

International Journal of Intellectual Advancements and Research in Engineering Computations

Enabling privacy-preserving shortest distance queries on encrypted data in cloud

[1]S.Rajeshwari, [2]K.Sharmila, [3]S.Thilothini,
[4]Mr.R.T.Dineshkumar, M. Tech.,

UG Scholar's, Department of Computer Science and Engineering, Vivekanandha College of Engineering For Women [Autonomous], Tiruchengode - 637 205, Tamilnadu, India.[1,2,3]

Assistant Professor, Department. of Computer Science and Engineering, Vivekanandha College of Engineering For Women [Autonomous], Tiruchengode - 637 205, Tamilnadu, India.[4]

ABSTRACT

Cloud storage provides a convenient, massive, and scalable storage at low cost, but data privacy is a major concern that prevents users from storing files on the cloud trustingly. One way of enhancing privacy from data owner point of view is to encrypt the files before outsourcing them onto the cloud and decrypt the files after downloading them. However, data encryption is a heavy overhead for the mobile devices, and data retrieval process incurs a complicated communication between the data user and cloud. Normally with limited bandwidth capacity and limited battery life, these issues introduce heavy overhead to computing and communication as well as a higher power consumption for mobile device users, which makes the encrypted search over mobile cloud very challenging .In this work, we propose BKCM (Binary Keyword Co-ordinate Matching), a bandwidth and energy efficient encrypted search architecture over mobile cloud. The proposed architecture offloads the computation from mobile devices to the cloud, and we further optimize the communication between the mobile clients and the cloud. It is demonstrated that the data privacy does not degrade when the performance enhancement methods are applied. Our experiments show that BKCM reduces the computation time by 23% to 46% and save the energy consumption by 35% to 55% per file retrieval, meanwhile the network traffics during the file retrievals are also significantly reduced.

Keywords: Binary Keyword Co-ordinate Matching, Cryptography Technique, Bandwidth

INTRODUCTION

Cloud Computing

Resource sharing in a pure plug and play model that dramatically simplifies infrastructure planning is the promise of „cloud computing". The two key advantages of this model are ease of use and cost-effectiveness. Though there remain questions on aspects such as security and vendor lock-in, the benefits this model offers are many. This work explores some of the basics of cloud computing with the aim of introducing aspects such as: Realities and risks of the model Components in the model Characteristics and Usage of the model This work aims to provide a means of understanding the model and exploring options available for complementing your technology and infrastructure needs.Cloud computing is a computing paradigm, where a large pool of systems are connected in private or public networks, to provide dynamically scalable infrastructure for application, data and file storage. With the advent of this technology, the cost of computation, application hosting, content storage and delivery is reduced significantly.

Cloud computing is a practical approach to experience direct cost benefits and it has the potential to transform a data center from a capital-intensive set up to a variable priced environment.The idea of cloud computing is based on a very fundamental principal of reusability of IT capabilities. The difference that cloud computing

Author for correspondence:
Department of Computer Science and Engineering, Vivekanandha College of Engineering For Women [Autonomous], Tiruchengode - 637 205, Tamilnadu, India.

brings compared to traditional concepts of “grid computing”, “distributed computing”, “utility computing”, or “autonomic computing” is to broaden horizons across organizational boundaries.

Public Cloud

Public clouds are owned and operated by third parties; they deliver superior economies of scale to customers, as the infrastructure costs are spread among a mix of users, giving each individual client an attractive low-cost, “Pay-as-you-go” model. All customers share the same infrastructure pool with limited configuration, security protections, and availability variances. One of the advantages of a public cloud is that they may be larger than an enterprises cloud, thus providing the ability to scale seamlessly, on demand.

Private Cloud

Private clouds are built exclusively for a single enterprise. They aim to address concerns on data security and offer greater control, which is typically lacking in a public cloud.

Hybrid Cloud

Hybrid Clouds combine both public and private cloud models. With a Hybrid Cloud, service providers can utilize 3rd party Cloud Providers in a full or partial manner thus increasing the flexibility of computing. The Hybrid cloud environment is capable of providing on-demand, externally provisioned scale. The ability to augment a private cloud with the resources of a public cloud can be used to manage any unexpected surges in workload.

Privacy of data: No one can uncover information about data content from the query and response as well as the cipher text itself

.

Privacy of the data owner: No one can learn about the identity of the data owner from the encrypted content. They suggested that one’s identity can be viewed as a combination of several attributes expressing the characteristics of the user in the form of access policy by using Boolean expressions such as AND, OR, or NOT. On the other hand, CP-ABE is complementary to KP-ABE by enabling encrypt or to specify access policy combined with the cipher text. Both schemes allow secure one-to-many communications such as targeted broadcasts for a specific group and individual user according to their attributes, some studies suggested modification of ABE schemes by hiding the access policy. These schemes operate on the assumption that the data owner directly delivers the cipher text to the receiver without an intermediate third party. In other words, when adopting those approaches directly in cloud storage, decryption keys can be exposed to an unauthorized third party. Hence, they are not feasible for data retrieval services in the cloud storage systems because the test procedure allows the CSP to learn which attributes the user has motivate and solve the problem of supporting effective ranked keyword search for providing efficient use of remotely stored encrypted data in Cloud. They firstly give a fundamental scheme and show that by same existing searchable encryption method, it is not efficient to achieve ranked search. So, they appropriately weaken the security guarantee, to solve this security problem they developed cryptography first OPSE, and derive an efficient one-to-many order-preserving mapping function, which allows the effective RSSE to be designed. Through thorough security analysis, they show that their proposed solution is secure and maintain the privacy, while correctly realizing the aim of ranked keyword search. And also shows that their solution enjoys “as-strong-as possible” security guarantee compared to conventional SSE schemes; Note that in their design, they focus on single keyword search. This section includes involved work and identified issues in system in addition to that an optimum solution is also provided. The cloud environment provides support for efficient computing and enables to provide the storage solutions at the remote end. The main aim is to address the following issues in the existing cloud storage:

1. **Data security:** The data is placed on the cloud which is not much secured due to third party access and treads therefore the data security in cloud storage is required
2. **Data owner and client privacy management:** The data owner and client in not distinguishable using the data additionally the privacy on such data is access is required.
3. **Searchable data space:** The cryptographic manner of data security converts the formats and not a bit of data recovered during the information retrieval. In this paper we discussed various method of searchable encryption to secure the data in cloud storage. Also, we discussed about cryptography methods which helps to convert the data from readable to unreadable form so our data could be saved in cloud storage from adversary. By studying all these papers, we can conclude that the essence of security for cloud storage is very necessary so that client could feel secure while accessing the cloud storage services. In this work we proposed a scheme for secure data accessing with maintaining its

privacy by using strong cryptographic algorithm. Our future work will attempt to enhance the feasible solution.

Literature Survey

"A Break In The Clouds: Towards A Cloud Definition,"

In this work L. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner et.al has proposed Cloud Computing is associated with a new paradigm for the provision of computing infrastructure [1]. This paradigm shifts the location of this infrastructure to the network reduce the costs associated with the management of hard- ware and software resources. The Cloud is drawing the attention from the Information and Communication Technology (ICT) community, thanks to the appearance of a set of services with common characteristics, provided by important industry players. Taking these features into account we can provide an encompassing definition of the Cloud. Obviously, the Cloud concept is still changing and these definitions show how the Cloud is conceived today: Clouds are a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services) [2]. These resources can be dynamically re- configured to adjust to a variable load, allowing also for optimum resource utilization. This pool of resources is typically exploited by a pay- per-use model in which guarantees are offered by the Infrastructure Provider by means of customized SLAs.

Virtualized In-Cloud Security Services For Mobile Devices

In this work J. Oberheide, K. Veeraraghavan, E. Cooke, J. Flinn, and F. Jahanian, et.al [3] has proposed Modern mobile devices continue to approach the capabilities and extensibility of standard desktop PCs. Unfortunately; these devices are also beginning to face many of the same security threats as desktops. Currently, mobile security solutions mirror the traditional desktop model in which they run detection services on the device. This approach is complex and resource intensive in both computation and power. This paper proposes a new model whereby mobile antivirus functionality is moved to an off-device network service employing multiple virtualized malware detection engines. Our argument is that it is possible to spend bandwidth resources to significantly reduce on-device CPU, memory, and power resources. We demonstrate how our in cloud model enhances mobile security and reduces on-device software complexity, while allowing for new services such as platform specific behavioral analysis engines. Our benchmark son Nokia's N800 and N95 mobile devices show that our mobile agent consumes an order of magnitude less CPU and memory while also consuming less power in common scenarios compared to existing on-device antivirus software. The second major component of the architecture is a network service responsible for file analysis. The task of the network service is to determine whether a file is malicious or unwanted. Unlike existing antivirus software that cannot run multiple detection engines on a single device due to technical conflicts and resource constraints, moving detection capabilities to a network service allows the use of multiple antivirus engines in parallel by hosting them in virtualized containers. That is, each candidate file is analyzed by multiple detection engines to determine whether a file is malicious or unwanted.

When Mobile Is Harder Than Fixed: Demystifying Security Challenges In Mobile Environments

In this work J. Oberheide and F. Jahanianet.A lhas proposed Modern mobile platforms are reinventing the mobile landscape [4]. These devices run commodity operating systems and have complete multi-protocol networking stacks, UI toolkits and other fully-featured libraries. While past mobile platforms had limited functionality and were relatively closed to third-party applications and user extensibility, new mobile platforms ship with complex Internet, pro-ductility, communication, and application suites and strongly encourage third party development with comprehensive soft-ware development kits and application delivery mechanisms. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. We expect consumer mobile platforms and devices to continue their rapid expansion in terms of sophistication and functionality. As their popularity increases and they be-come enticing targets for attackers, these devices will face a range of new security threats. We believe that domain of mobile security presents a number of interesting challenges that are becoming ever-important to explore as the adoption and use of these mobile platforms continues to accelerate.

Managing Gigabytes: Compressing And Indexing Documents And Images.

In this work A. A. Moffat, T. C. Bell et.al has proposed the book is a treasure trove of techniques for compressing and indexing [5]. It will be handy for those interested in indexing and retrieving information from large text databases running to several gigabytes. The style of presentation is laudable and the coverage of topics is impressive. The authors could have included audio and video compression techniques. This good book will be very useful for the intended audience. The authors have incorporated many changes in the second edition by updating various chapters to include the latest developments. The book is concerned with the task of managing large volumes of text and image data amounting several gigabytes. The fundamental problems addressed in the book include compressing such data and indexing it, I order to enable easy search. Many books look at compressing and indexing as if they are unrelated techniques; however, this book brings out the advantages of combining them in a beneficial way. The book has been authored by academics and is therefore well suited for academic use. It may be used for teaching courses in the area of data compression and information retrieval at various levels. An instructor’s supplement is also available and includes test questions and review material for use during teaching.

“Practical Techniques For Searches On Encrypted Data”

In this work D. Song, D. Wagner, and A. Perrig, et.al has proposed Today’s mail servers such as IMAP servers, fileservers and other data storage servers typically must be fully trusted—they have access to the data, and hence must be trusted not to reveal it without authorization—which introduces undesirable security and privacy risks in applications [6]. Previous work shows how to build encrypted file systems and secure mail servers, but typically one must sacrifice functionality to ensure security. The fundamental problem is that moving the computation to the data storage seems very difficult when the data is encrypted, and many computation problems over encrypted data previously had no practical solutions. In this paper, we show how to support searching functionality without any loss of data confidentiality. An example is where a mobile user with limited bandwidth wants to retrieve all email containing the word “Urgent” from a untrusted mail-storage server in the infrastructure.In this paper, we describe our cryptographic schemes for the problem of searching on encrypted data and provide proofs of security for there sulting crypto systems. Our techniques have a number of crucial advantages. They are provably secure: they provide provable secrecy for encryption, in the sense that the untrusted server cannot learn anything about the plaintext when only given the ciphertexts; they provide query isolation for searches, meaning that the untrusted server cannot learn anything more aboutthe plaintext than the search result; they provide controlled searching, so that the untrusted server cannot search for anarbitrary word without the user’s authorization;

“Public Key Encryption With Keyword Search,”

In this work D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano et.al has proposed Suppose user Alice wishes to read her email on a number of devices: laptop, desktop, pager, etc.Alice's mail gateway is supposed to route email to the appropriate device based on the keywords in the email. For example, when Bob sends email with the keyword \urgent" the mail is routed to Alice's pager [7]. When Bob sends email with the keyword \lunch" the mail is routed to Alice's desktop for reading later. One expects each email to contain a small number of keywords. For example,all words on the subject line as well as the sender's email address could be used as keywords. The mobile people project provides this email processing capability. Now, suppose Bob sends encrypted email to Alice using Alice's public key. Both the contents of the email and the keywords are encrypted. In this case the mail gateway cannot see the keywords and hence cannot make routing decisions. As a result, the mobile people project is unable to process secure email without violating user privacy. Our goal is to enable Alice to give the gateway the ability to test whether \urgent" is a keyword in the email, but the gateway should learn nothing else about the email. More generally, Alice should be able to specify a few keywords that the mail gate way can search for, but learn nothing else about incoming mail.

“Searchable Symmetric Encryption: Improved Definitions And Efficient Constructions,”

In this work R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky et.al has proposed Searchable symmetric encryption (SSE) allows a party to outsource the storage of his data to another party in a private manner, while maintaining the ability to selectively search over It [8]. This problem has been the focus of active research and several security denations and constructions have been proposed. In this paper we begin by reviewing existing notions of security and proposenew and stronger security denations. We then pre sent two constructions that we show secure under our new denations. Interestingly, in addition to satisfying stronger security guarantees, our constructions are more efficient than all previous constructions. Further, prior work on SSE only considered the setting where only the owner of the data is capable of submitting search queries. We consider the natural extension where an arbitrary group of parties other than the owner can submit search queries. We formally dene SSE in this multi-user setting, and present an efficient construction. In this article, we have revisited the problem of searchable symmetric encryption, which allows a client to store its data on a remote server in such a way that it can search over it in a private manner.We make several contributions including new security definitions and new constructions [9]. Motivated by subtle problems in all previous security definitions for SSE, we propose new denations and point out that the existing notions have significant practical drawbacks: contrary to the natural use of searchable encryption, they only guarantee security for users that perform all their searches at once [10].

Existing System

To effectively support an encrypted search scheme with high security level over cloud data, we introduce a new architecture that we name BKCM. Our aim is to design a practical solution for secure encrypted search over a mobile cloud storage. We first introduce the design idea and then introduce development of our own protocol with the change of the traditional process of file search and retrieval for the cloud data Our scheme achieves the security and efficiency goals mentioned above. Thereafter, we discuss the reasons why BKCM can achieve performance efficiency.

The Basic Idea of BKCM

The basic idea behind BKCM is to offload the calculation and the ranking load of the relevance scores to the cloud. It has been highlighted that offloading some computation intensive applications onto the cloud can be an efficient low power design philosophy [12]. Cloud providers can provide computing cycles, and users can use these cycles to reduce the amounts of computation on mobile systems and save energy. However, at the same time, offloaded applications intend to increase the transmission amount and thus increase the energy consumption from another aspect [13]. This double effect motivates us to carefully redesign the traditional file encrypted search and retrieval process [14]. We first take an overview of major processes for all file encrypted search and retrieval schemes. There are normally three main processes:

- The process of authentication is used by the data owner to authenticate the data users.
- The file set and its index are stored in the cloud after being encrypted by the data owner during the preprocessing and indexing stages.
- The data user searches the files corresponding to a keyword by sending a request to the cloud server in the search and retrieval processes.

We now introduce the detailed design how BKCM addresses the power efficiency and the security challenges in modifying these processes [15].

Disadvantages

- Less accuracy.
- Insufficient data access may occur.
- Poor performance.
- High in computation time.
- Less accuracy and waste on mobile data energy.

Proposed System

Bkcm Implementation For Security Enhancement For Mobile Cloud

In order to achieve security enhancement with energy and traffic efficiency, we implement the modules in BKCM using modified routines and new algorithms. Our system will be introduced in three parts. As previously mentioned, the data owner should build a TF table as index and encrypt it using OPE in order to offload the calculation and ranking load of the relevance scores to the cloud. So as to control the statistics information leak, we implement our one-to-many OPE in the data owner module. We also wrap the keywords to be searched by adding some noise in the data user module to help controlling the keywords-files association leak. In order to get top-k relevant files, we implement a ranking function to calculate the relevant score on the cloud. Given a keyword in ORS, the cloud server is in charge of calculating the relevance scores for the data user to get the corresponding top-k relevant files. Therefore, we implement both the unwrap and rank functions in the cloud server module. Hence these modules are modified compared with the traditional ones.

Redesign of the Data Owner Module

We modify the way of building the index to support the ORS scheme by our one-to-many OPE and implement it to control the statistics information leak. The authentication between the data owner and the data user is also redesigned in order to ensure the security of BKCM. We now elaborate the implementation of the index construction, the encryption functions and detail the authentication process.

TF-IDF:TF-IDFis the product of two statistics,term frequency and inverse document frequency. Various ways for determining the exact values of both statistics exist. In the case of the term frequency (TF)tf(t,d),the simplest choice is to use the raw frequency of a term in

a document, i.e. the number of times that term t occurs in documented. If we denote the raw frequency of tbyf(t, d), then the simple TF scheme is t f (t, d) = f(t, d).

Security Analysis And Evaluation

We analyse the security of BKCM based on important security threats. Concretely, the most important principle of the design is to prevent the attacker from obtaining any plaintext information regarding our data file set or the searched keyword. Then we should let the trusted but curious mobile cloud server learn as little information as possible. Last but not least, an unauthenticated data users should not be able to perform any file retrieval. We ran experiments to test the security of BKCM.

Server Information Acquisition Control

We assume that the cloud storage system provider will not collude with malicious users or intrude users' data intentionally. The cloud server can infer and analyze the encrypted index and get additional information, but it has no intention to modify any important data. We use the private cloud server from our school and assumed it as honest and perform important calculations here. This assumption is also used in most of the previous work.

In BKCM, the cloud server calculates the relevance score and finds the most relevant files corresponding to a given keyword. In order to find the TF value of a queried keyword, the cloud server should also know the unwrap function of the user-supplied wrapped tuple. Therefore, the cloud server gets more information than any potential attacker. Therefore, a curious cloud server may determine the terms queried by the users only by comparing the queries and the results. As most of the previous schemes, we assumed that our test cloud server semi-trusted, and therefore we only need to minimize the amount of information it acquires. Moreover, in terms of performance improvements, information leakage does not seem to be a very serious problem, and updating theTF table periodically also protects the index from being inferred by the server.

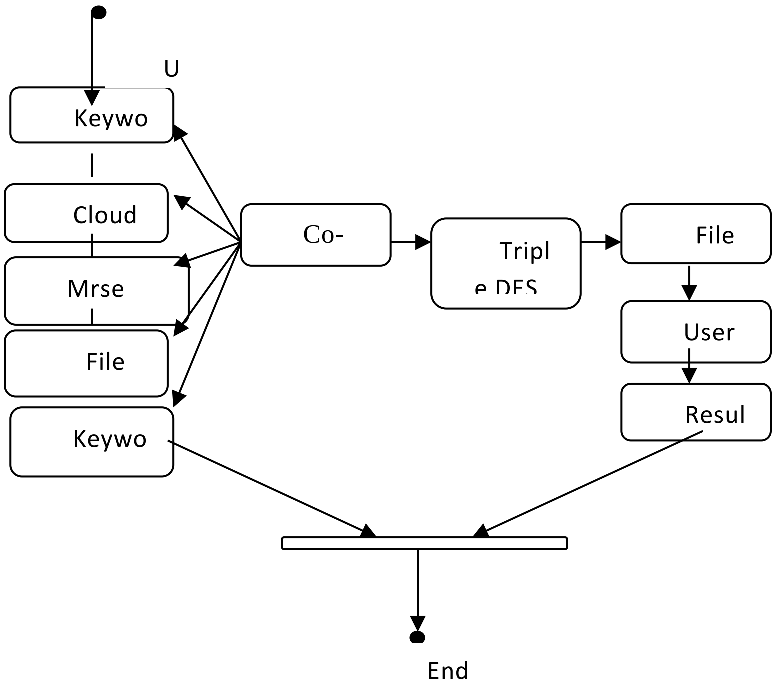
Note that BKCM is established with widely used TFIDF encryption approach, and all nature defects of this encrypted search scheme cannot be completely resolved even BKCM. In addition, when a data user performs a search, the keyword to be queried will be encrypted by the data owner's key. The user receives a hash table after the first time it is authorized by the data owner.In order to prevent these users from running secret searches maliciously.

Advantages

- "Coordinate matching" by inner keyword similarity is possible.
- Secured Multi keyword ranked ontology keyword mapping and search: To design search schemes which allow multi-keyword query and provide result similarity ranking for valuable data retrieval, instead of returning undifferentiated results.

Privacy: To prevent cloud server from learning additional information from dataset and index, and to meet privacy requirements. Effectiveness with high performance: Above goals on functionality and privacy should be achieved with low communication and computation overhead

System Model



Methodology

Modules

- Encrypt Module/Client Module
- Multi-keyword Ontology mapping Module
- Admin Module/File upload Module
- Ranking Result

Encrypt Module/Client Module

This module is used to help the server to encrypt the document using TRIPLE DES Algorithm and to convert the encrypted document to the Zip file with activation code and then activation code sends to the user for download. This module is used to help the client to search the file using the multiple key words concept and get the accurate result list based on the user query. The user is going to select the required file and register the user details and get activation code in mail from the "customerservice404" email before enter the activation code. After user can download the Zip file and extract that file.

Multi-keyword Ontology mapping Module

This module is used to help the user to get the accurate result based on the multiple keyword concepts. The users can enter the multiple words query, the server is going to split that query into a single word after search that word file in our database. Finally, display the matched word list from the database and the user gets the file from that list However, directly outsourcing data vector or query vector will violate index privacy or search privacy. To meet the challenge of supporting such multi-keyword semantic without privacy breaches, we propose a basic SMS scheme using secure inner product computation, which is adapted from a secure k-nearest neighbor (kNN) technique.

Admin Module/File Upload Module

This module is used to help the server to view details and upload files with the security. Admin uses the log key to the login time. Before the admin logout, change the log key. The admin can change the password after the login and view the user downloading details and the counting of file request details on flowchart. The admin can upload the file after the conversion of the Zip file format.

Ranking Result

When any User request for the data then Ranking is done on requested data using k-nearest neighbor algorithm. For Ranking co-ordinate matching‖ principle is used. After ranking user gets the expected results of the query.

EXPERIMENTAL RESULTS

In our experiments, we use a data set of 1000 files with different sizes and a VM in the cloud with Dual vCPUs at2.27GHz. An android smart phone with a CPU at 1GHz sends the queries as the mobile client of BKCM through an about 8M wireless network. An Android program receives the user's input and encrypts it before getting the hash value and then wrap it into a tuple which is sent to the mobile cloud server. Another feature of this program is to retrieve the files back from the mobile cloud server and decrypt them. In addition, we implemented both TRS and PTS for a comparative purpose. As energy consumption is critical for mobile devices, we evaluate BKCM energy efficiency in this subsection. We use a phone power monitor to accurately measure the system energy consumption. Although slight changes depending upon the environment might occur, the comparison is quite accurate as controlled trials were performed. Observe that the energy consumption is reduced from0.08mAh to 0.036mAh when searching and retrieving files of size 100KB, which means that ORS saves 55%energy compared to TRS. When searching and retrieving files of 1MB size, the energy consumption is reduced from 0.164mAh to 0.106mAh, that means a 35% energy saving. So, BKCM provides a very efficient power consumption. For example, to exhaust our 1650mAh battery,ORS (of BKCM) can perform 22000 retrievals while TRS could only retrieve 13000 files of size 600KB.

File Search And Retrieval Time

We compare the File Searching and Retrieval Time (FSRT) for the three schemes in this subsection as illustrated in Figure 10. We test the FSRT for different files with size ranging from 100KB to 1MB. We observe that the FSRT of PTS is the shortest since it does not have to perform any security computation. The FSRT of ORS is effectively reduced when compared to the one of TRS.This difference is due to the advantages of the BKCM design in terms of relevance score calculation offloading, and thus leads to reduction of file search and retrieval process. The FSRT value of ORS is very near to the one of PTS, implying a very low cost to security on the mobile device. For example, BKCM saves FSRT by 46% compared to TRS for files of size 100KB, and by 23% for 1MB files.The file retrieval time only depends on the file size and network bandwidth. When offered a greater bandwidth, BKCM becomes more efficient since downloading time of files becomes a bottleneck of other schemes. The decryption time of the files is equal in all schemes and it is therefore pointless to measure it.

	PTS	**TRS**	**ORS**
Request/Response	190ms	370ms	190ms
Stemming and Encryption	0	10ms	10ms
Hash and Wrap	0	145ms	150ms
Server file search	80ms	70ms	75ms
Client file search	0	260ms	0
Sum	270ms	855ms	425ms

Fsrt Analyse Of Pts, Trs And Ors

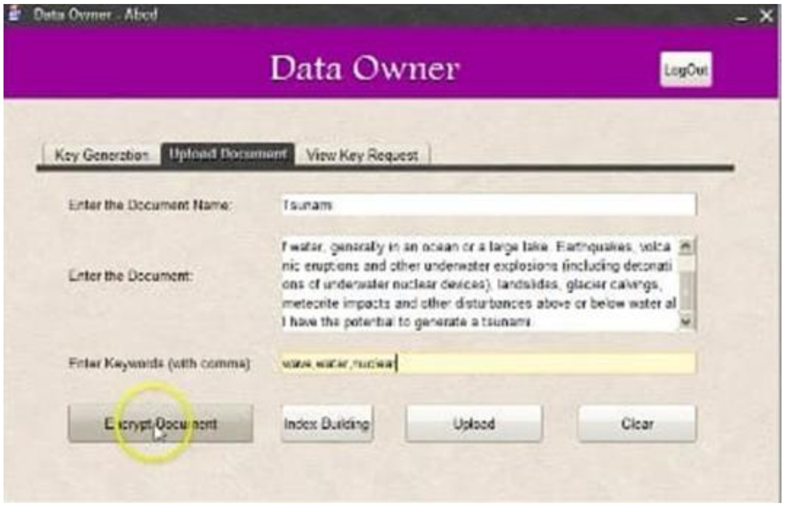
The efficient FSRT of BKCM is achieved by improving the process efficiency, since only a single round of communication and relevance score calculation offload is used. The searching process is analyzed in Table 1. Without any security service, PTS (Plain Text search) does not spend any time on stemming and encryption; neither does it on hash and wrap. On the other hand,ORS and TRS provide encrypted search schemes with related overhead. As shown in Table 1, ORS can improve the “request/response” time significantly than TRS from 370ms to 190ms (saving 180ms), and eliminate the “client file search” time by offloading it onto the server (saving260ms). Notice that the “sever file search” calculation workload of ORS is 75ms, which is 5ms longer than that of TRS. This is explained by the fact that the server takes the offloaded search calculation of the mobile user. In the other words, BKCM eliminates the “client file search” time at the cost of a little heavier “server file search” time. This proves that the offloading is highly efficient (5ms vs.260ms). Moreover, ORS spends more 5ms on wrapping the hash value than TRS for enhancing the security. Note that the “server file search” time of PTS is higher than the other two schemes, since the server should execute stem and hash function for plaintext file search, while the hash functions are executed by the mobile data user in both TRS and ORS. Overall, ORS is secure and effective.

Throughput

The calculation offload from the mobile device to the cloud data center reduces the execution time of the relevance score calculation due to the higher server capacity. Therefore, this also greatly increases the system throughput besides FSRT improvement; We observe that the file access acceleration is very effective when dealing with small files as the relevance score calculation is executed more frequently. For example, on a 100KB file, the access speed is increased from 104KB/s to 194KB/s, almost doubling the throughput. The acceleration is still effective when accessing files with size 1MB (29.6% acceleration). The throughput of ORS is not much less than that of PTS.

Implementation to achieve an encrypted search in a mobile cloud. The security study of BKCM showed that it is secure enough for mobile cloud computing, while a series of experiments highlighted its efficiency. BKCM is slightly more time and energy consuming than keyword search over plain-text, but at the same time it saves significant energy compared to traditional strategies featuring a similar security level. Based on BKCM, this work can be extended to more other novel implementations. We have proposed a single keyword search scheme to make encrypted data search efficient. However, there are still some possible extensions of our current work remaining. We would like to propose a multi-keyword search scheme to perform encrypted data search over mobile cloud in future. As our OPE algorithm is a simple one, another extension is to find a powerful algorithm which will not harm the efficiency.

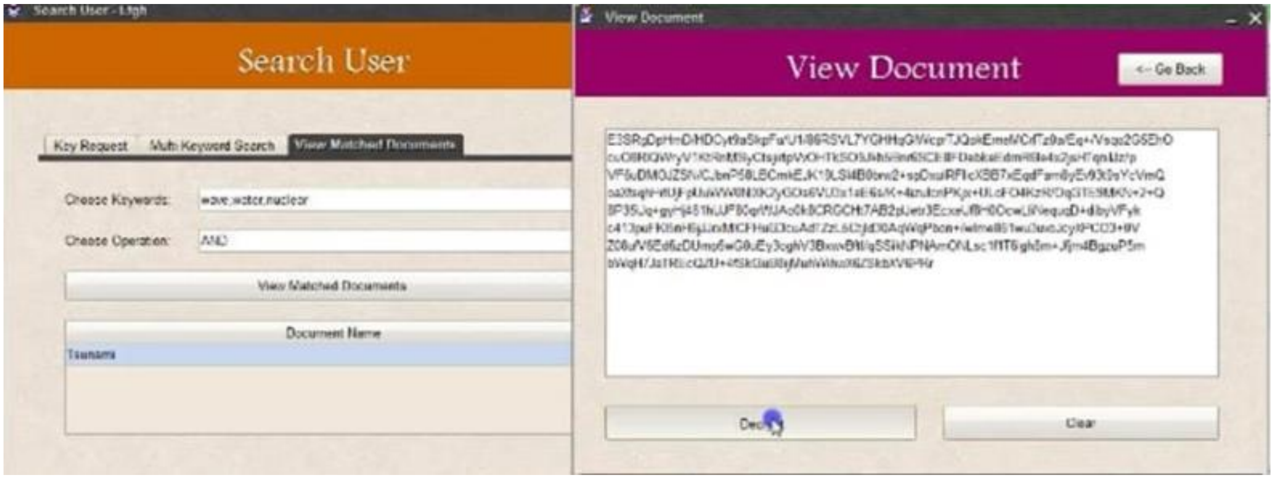
RESULTS



Enter the Document Name: Tsunami

Enter the Document: f water, generally in an ocean or a large lake. Earthquakes, volcanic eruptions and other underwater explosions (including detonations of underwater nuclear devices), landslides, glacier calvings, meteorite impacts and other disturbances above or below water all have the potential to generate a tsunami

Enter Keywords (with comma): wave,water,nuclear



Choose Keywords: wave,water,nuclear

Choose Operation: AND

Document Name
Tsunami

[illegible]

CONCLUSION

we developed a new architecture, BKCMasan initial attempt to create a traffic and energy efficient encrypted keyword search tool over mobile cloud storages. We started with the introduction of a basic scheme that we compared to previous encrypted search tools for cloud computing and showed their inefficiency in mobile cloud context. Then we developed an efficient implementation to achieve an encrypted search in a mobile cloud. The security study of BKCM showed that it is secure enough for mobile cloud computing, while a series of experiments highlighted its efficiency. BKCM is slightly more time and energy consuming than keyword search over plain-text, but at the same time it saves significant energy compared to traditional strategies featuring a similar security level. Based on BKCM, this work can be extended to more other novel implementations. We have proposed a single keyword search scheme to make encrypted data search efficient. However, there are still some possible extensions of our current work remaining. We would like to propose a multi-keyword search scheme to perform encrypted data search over mobile cloud in future. As our OPE algorithm is a simple one, another extension is to find a powerful algorithm which will not harm the efficiency.

REFERENCES

[1]. L. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, “A break in the clouds: towards a cloud definition,” ACM SIGCOMM Computer Communication Review, vol. 39, no. 1, pp. 50–55, 2008.

[2]. D. Huang, “Mobile cloud computing,” IEEE COMSOC Multimedia Communications Technical Committee (MMTC) E-Letter, 2011.

[3]. J. Oberheide, K. Veeraraghavan, E. Cooke, J. Flinn, and F. Jahanian, “Virtualized in-cloud security services for mobile devices,” in Proceedings of the First Workshop on Virtualization in Mobile Computing. ACM, 2008, pp. 31–35.

[4]. J. Oberheide and F. Jahanian, “When mobile is harder than fixed (and vice versa): demystifying security challenges in mobile environments,” in Proceedings of the Eleventh Workshop on Mobile Computing Systems ACM, 2010, 43 48.

[5]. A.Moffat, T. C. Bell et al., Managing gigabytes: compressing and indexing documents and images. Morgan Kaufmann Pub, 1999.

[6]. D. Song, D. Wagner, and A. Perrig, “Practical techniques for searches on encrypted data,” in Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on. IEEE, 2000, pp. 44– 55.

[7]. D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, “Public key encryption with keyword search,” in Advances in Cryptology- Eurocrypt 2004. Springer, 2004, pp. 506–522.

[8]. R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, “Searchable symmetric encryption: improved definitions and efficient constructions,” in Proceedings of the 13th ACM conference on Computer and communications security. ACM, 2006, pp. 79–88.

[9]. Y. Chang and M. Mitzenmacher, “Privacy preserving keyword searches on remote encrypted data,” in Applied Cryptography and Network Security. Springer, 2005, pp. 391–421.

[10]. S. Zerr, D. Olmedilla, W. Nejdl, and W. Siberski, “Zerber+ r: Topk retrieval from a confidential index,” in Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology. ACM, 2009, pp. 439–449.

[11]. C. Wang, N. Cao, K. Ren, and W. Lou, “Enabling secure and efficient ranked keyword search over outsourced cloud data,” Parallel and Distributed Systems, IEEE Transactions on, vol. 23, no. 8, pp. 1467–1479, 2012.

[12]. N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, “Privacy-preserving multi-keyword ranked search over encrypted cloud data,” Parallel and Distributed Systems, IEEE Transactions on, vol. 25, no. 1, pp. 222–233, 2014.

[13]. J. Zobel and A. Moffat, “Inverted files for text search engines,” ACM Computing Surveys (CSUR), vol. 38, no. 2, p. 6, 2006.

[14]. R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, “Order preserving encryption for numeric data,” in Proceedings of the 2004 ACM SIGMOD international conference on Management of data. ACM, 2004, pp. 563–574.

[15]. D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” the Journal of machine Learning research, vol. 3, pp. 993–1022, 2003.