A Novel Feature Matching Rank Search Mechanism over Encrypted Cloud Data

Mrunal Jaysing Jadhav [1], Snehal Sandip Navathar [2], Rajeshwari Chandrakanth Chavan [3], Prof Ashwini Jadhav[4]

Genba Sopanrao Moze College of Engineering Pune

Abstract: A Secure and Dynamic Multi-keyword Ranked Search Scheme over Encrypted Cloud Data Due to the increasing popularity of cloud computing, more and more data owners are motivated to outsource their data to cloud servers for great convenience and reduced cost in data management. However, sensitive data should be encrypted before outsourcing for privacy requirements, which obsoletes data utilization like keyword-based document retrieval. In this paper, we present a secure multi-keyword ranked search scheme over encrypted cloud data, which simultaneously supports dynamic update operations like deletion and insertion of documents. Specifically, the vector space model and the widely-used TFIDF model are combined in the index construction and query generation. We construct a special tree-based index structure and propose a "Greedy Depth-first Search" algorithm to provide efficient multi-keyword ranked search. The secure kNN algorithm is utilized to encrypt the index and query vectors, and meanwhile ensure accurate relevance score calculation between encrypted index and query vectors. In order to resist statistical attacks, phantom terms are added to the index vector for blinding search results. Due to the use of our special tree-based index structure, the proposed scheme can achieve sub-linear search time and deal with the deletion and insertion of documents flexibly. Extensive experiments are conducted to demonstrate the efficiency of the proposed scheme.

Technical Keywords: Searchable encryption, multi-keyword ranked search, dynamic update, cloud computing.

1. INTRODUCTION

Cloud computing has been considered as another model of big business IT base, which can sort out gigantic asset of processing, stockpiling and applications, and empower clients to appreciate pervasive, advantageous and on-interest system access to a common pool of configurable registering assets with extraordinary efficiency and negligible monetary overhead. Pulled in by these engaging components, both people and undertakings are spurred to outsource their information to the cloud, rather than acquiring programming and equipment to deal with the information themselves. In spite of the different points of interest of cloud administrations, outsourcing touchy data, (for example, messages, individual wellbeing records, organization finance information, government archives, and so on.) to remote servers brings protection concerns. The cloud administration suppliers (CSPs) that keep the information for clients may get to clients' touchy data without approval. A general way to deal with secure the information confidentiality is to encode the information before outsourcing. Be that as it may, this will bring about an immense expense regarding information ease of use. For instance, the current systems on magic word based data.

2. LITERATURE SURVEY

1. Security and Privacy Issues in C Cloud Computing.

AUTHORS: Jaydip Sen Cloud computing transforms the way information technology (IT) is consumed and managed, promising improved cost efficiencies, accelerated innovation, faster time-to-market, and the ability to scale applications on demand (Leighton, 2009). According to Gartner, while the hype grew exponentially during 2008 and continued since, it is clear that there is a major shift towards the cloud computing model and that the benefits may be substantial (Gartner Hype-Cycle, 2012). However, as the shape of the cloud computing is emerging and developing rapidly both conceptually and in reality, the legal/contractual, economic, service quality, interoperability, security and privacy issues still pose significant challenges. In this chapter, describe various service and deployment models of cloud computing and identify major challenges.

2. Cryptographic Cloud Storage

AUTHORS: Seny Kamara

Author considers the problem of building a secure cloud storage service on top of a public cloud infrastructure where the service provider is not completely trusted by the customer. Here describe, at a high level, several architectures that combine recent and non-standard cryptographic primitives in order to achieve goal. Authors survey the benefits such an architecture would provide to both customers and service providers and give an overview of recent advances in cryptography motivated specifically by cloud storage.

3. A fully homeopathic encryption scheme

AUTHORS: Craig Gentry Author propose the first fully holomorphic encryption scheme, solving a central open problem in cryptography. Such a scheme allows one to compute arbitrary functions over encrypted data without the decryption key – i.e., given encryptions E(m1),...,E(mt) of m1,...,mt, one can efficiently compute a compact cipher text that encrypts f(m1,...,mt) for any efficiently computable function f. This problem was posed by Rivets et al. in 1978. Fully holomorphic encryption has numerous applications. For example, it enables private queries to a search engine – the user submits an encrypted query and the search engine computes a succinct encrypted answer without ever looking at the query in the clear. It also enables searching on encrypted data – a user stores encrypted files on a remote file server and can later have the server retrieve only files that (when decrypted) satisfy some Boolean constraint, even though the server cannot decrypt the files on its own. More broadly, fully holomorphic encryption improves the efficiency of secure multiparty computation.

3. Public Key Encryption with keyword Search

AUTHORS: Dan Boneh Author studies the problem of searching on data that is encrypted using a public key system. Consider user Bob who sends email to user Alice encrypted under Alice's public key. An email gateway wants to test whether the email contains the keyword "urgent" so that it could route the email accordingly. Alice, on the other hand does not wish to give the gateway the ability to decrypt all her messages. Here define and construct a mechanism that enables Alice to provide a key to the gateway that enables the gateway to test whether the word "urgent" is a keyword in the email without learning anything else about the email. Here refer to this mechanism as Public Key Encryption with keyword Search. As another example, consider a mail server that stores various messages publicly encrypted for Alice by others. Using this mechanism Alice can send the mail server a key that will enable the server to identify all messages containing some specific keyword, but learn nothing else. Here define the concept of public key encryption with keyword search and give several constructions.

4. Public Key Encryption That Allows PIR Queries

AUTHORS: Dan Boneh Consider the following problem: Alice wishes to maintain her email using a storage-provider Bob (such as a Yahoo! or Hotmail e-mail account). This storage-provider should provide for Alice the ability to collect, retrieve, search and delete emails but, at the same time, should learn neither the content of messages sent from the senders to Alice (with Bob as an intermediary), nor the search criteria used by Alice. A trivial solution is that messages will be sent to Bob in encrypted form and Alice, whenever she wants to search for some message, will ask Bob to send her a copy of the entire database of encrypted emails. This however is highly inefficient. Communication efficient and at the same time, respect the privacy of Alice. In this paper, show how to create a public-key encryption scheme for Alice that allows PIR searching over encrypted documents. This solution is the first to reveal no partial information regarding the user's search (including the access pattern) in the public-key setting and with non-trivially small communication complexity. This provides a theoretical solution to a problem posed by Boneh, DiCrescenzo, Ostrovsky and Persiano on "Public-key Encryption with Keyword Search." The main technique of our solution also allows for Single-Database PIR writing with sub-linear communication complexity, which we consider of independent interest.

5. Practical Techniques for Searches on Encrypted Data

AUTHORS: Dawn Xiaodong Song

It is desirable to store data on data storage servers such as mail servers and file servers in encrypted form to reduce security and privacy risks. But this usually implies that one has to sacrifice functionality for security. For example, if a client wishes to retrieve only documents containing certain words, it was not previously known how to let the data storage server perform the search and answer the query without loss of data confidentiality. In this paper, describe cryptographic schemes for the problem of searching on encrypted data and provide proofs of security for the resulting crypto systems. This technique has 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 cipher text; they provide query isolation for searches, meaning that the untrusted server cannot learn anything more about the plaintext than the search result; they provide controlled searching, so that the untrusted server cannot search for an arbitrary word without the user's authorization; they also support hidden queries, so that the user may ask the untrusted server to search for a secret word without revealing the word to the server. The algorithms present are simple, fast and introduce almost no space and communication overhead, and hence are practical to use today.

3. PROBLEM DEFINITION

Sensitive data should be encrypted before outsourcing for privacy requirements, which obsoletes data utilization like keyword-based document retrieval. Distributed computing has been considered as another model of big business IT base, which can sort out gigantic asset of processing, stockpiling and applications, and empower clients to appreciate pervasive, advantageous and on-interest system access to a common pool of configurable registering assets with extraordinary efficiency and negligible monetary overhead. Pulled in by these engaging components, both people and undertakings are spurred to outsource their information to the cloud, rather than acquiring programming and equipment to deal with the information themselves. In spite of the different points of interest of cloud administrations, outsourcing touchy data, (for example, messages, individual wellbeing records, organization finance information, government archives, and so on.) to remote servers brings protection concerns. The cloud administration suppliers (CSPs) that keep the information for clients may get to clients' touchy data without approval. A general way to deal with secure the information confidentiality is to encode the information before outsourcing. Be that as it may, this will bring about an immense expense regarding information ease of use. For instance, the current systems on magic word based data.

4. SYSTEM DESIGN

This system consists of three parts, that is, data owner, data user, and cloud server. The data owner first regards all kinds of data as documents, then extracts keywords from each document to create an encrypted index, and finally uploads the encrypted documents and indexes to the cloud server. While querying, the data user first generates a trapdoor through search control operation, and then submits it to the cloud server for retrieval. After receiving the trapdoor, the cloud server calculates the similarity score of the trapdoor and each document index, and then returns copies of encrypted documents to the data user according the score. In the end, the data user receives the encrypted results and decrypts them through access control operation

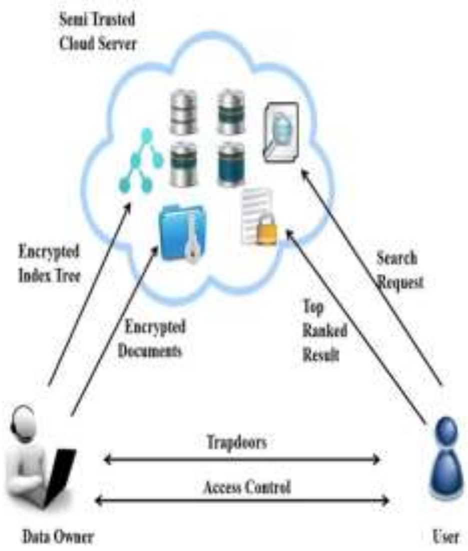


Fig.1 System Architecture

5. PROPOSED METHOD

5.1 Algorithm:

5.1.1 Algorithm 1: Build IndexTree(F):

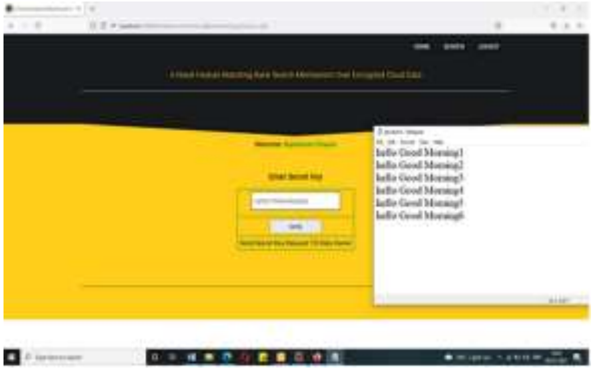
- Input: the document collection F = {f1, f2, FN} with the identifiers FID = {FID|FID = 1, 2, n}.
- Output: the index tree T

```
for each document fFID in F do
    Construct a leaf node u for fFID, with u.ID = GenID(), u.Pl = u.Pr = null, u.FID = FID, and D[i] = TFfFID,wi for i = 1,...,m;
    Insert u to CurrentNodeSet;
end for
while the number of nodes in CurrentNodeSet is larger than 1 do
    if the number of nodes in CurrentNodeSet is even, i.e. 2h then
        for each pair of nodes u′ and u″ in CurrentNodeSet do
            Generate a parent node u for u′ and u″, with u.ID = GenID(), u.Pl = u′, u.Pr = u″, u.FID = 0 and D[i] = max{u′.D[i],u″.D[i]} for each i = 1,...,m;
            Insert u to TempNodeSet;
        end for
    else
        for each pair of nodes u′ and u″ of the former (2h−2) nodes in CurrentNodeSet do
            Generate a parent node u for u′ and u″;
            Insert u to TempNodeSet;
        end for
        Create a parent node u1 for the (2h−1)-th and 2h-th node, and then create a parent node u for u1 and the (2h + 1)-th node;
        Insert u to TempNodeSet;
    end if
    Replace CurrentNodeSet with TempNodeSet and then clear TempNodeSet;
end while
return the only node left in CurrentNodeSet, name- ly, the root of index tree T;
```

5.1.2 Algorithm 2 GDFS(IndexTreeNode

```
if the node u is not a leaf node then
    if RScore(Du,Q) > kthscore then
        GDFS(u.hchild);
        GDFS(u.lchild);
    else
        return
    end if
else
    if RScore (Du,Q) > kthscore then
        Delete the element with the smallest relevance score from RList;
        Insert a new element⟨RScore (Du, Q), u.FID⟩and sort all the elements of RList;
    end if
    return
end if
```

6. RESULTS



hello Good Morning1
hello Good Morning2
hello Good Morning3
hello Good Morning4
hello Good Morning5
hello Good Morning6

Screenshot 1

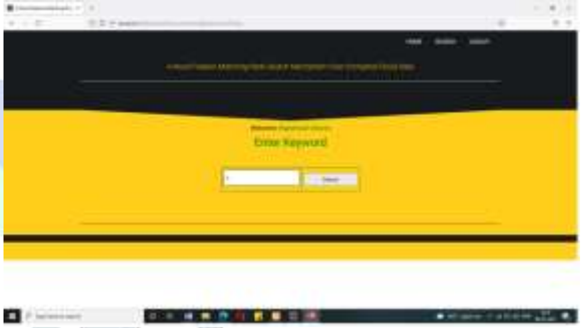
Screenshot 2

Screenshot 3

Screenshot 4

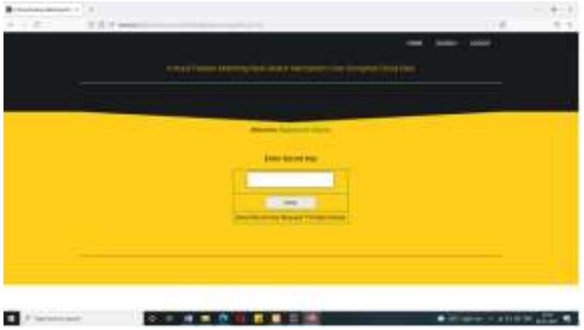
Screenshot 5

Screenshot 6

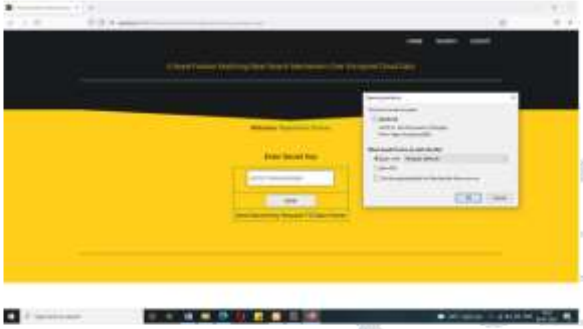


Screenshot 7

Screenshot 8



Screenshot 9



Screenshot 10

7. CONCLUSION

A protected, efficient and element inquiry plan is proposed, which bolsters the exact multi-essential word positioned hunt as well as the dynamic erasure and insertion of archives. We build an extraordinary decisive word adjusted parallel tree as the record, and propose a "Greedy Depth-first Search calculation to acquire preferred efficiency over straight pursuit. Also, the parallel inquiry procedure can be completed to further diminish the time cost. The plan's security is ensured against two danger models by utilizing the protected kNN calculation. Trial results show the efficiency of our proposed plan. There are still numerous test issues in symmetric SE plans. In the proposed plan, the information proprietor is in charge of producing overhauling data and sending them to the cloud server. Therefore, the information proprietor needs to store the decoded record tree and the data that are important to recalculate the IDF values. Such a dynamic information proprietor may not be exceptionally suitable for the distributed computing model. It could be a significant however difficult future work to plan an element searchable encryption plot who's overhauling operation can be finished by cloud server just, in the interim holding the capacity to bolster multi-pivotal word positioned pursuit. What's more, as the large portion of works about searchable encryption, our plan essentially considers the test from the cloud server. Really, there are numerous safe difficulties in a multi-client plan. Firstly, every one of the clients more often than not keep the same secure key for trapdoor era in a symmetric SE plan. For this situation, the client's disavowal is error.

8. FUTURE SCOPE

There are still many challenge problems in symmetric SE schemes. In the proposed scheme, the data owner is responsible for generating updating information and sending them to the cloud server. Thus, the data owner needs to store the unencrypted index tree and the information that are necessary to recalculate the IDF values. Such an active data owner may not be very suitable for the cloud computing model. It could be a meaningful but difficult future work to design a dynamic searchable encryption scheme whose updating operation can be completed by cloud server only, meanwhile reserving the ability to support multi-keyword ranked search. In addition, as the most of works about searchable encryption, our scheme mainly considers the challenge from the cl**oud server.**

9. REFERENCES

[1] K. Ren, C.Wang, Q.Wang et al., “Security challenges for the public cloud,” IEEE Internet Computing, vol. 16, no. 1, pp. 69–73, 2012.

[2] S. Kamara and K. Lauter, “Cryptographic cloud storage,” in Financial Cryptography and Data Security. Springer, 2010, pp. 136– 149.

[3] C. Gentry, “A fully homomorphic encryption scheme,” Ph.D. dissertation, Stanford University, 2009.

[4] 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.

[5] D. Boneh, E. Kushilevitz, R. Ostrovsky, and W. E. Skeith III, “Public key encryption that allows pir queries,” in Advances in Cryptology-CRYPTO 2007. Springer, 2007, pp. 50–67.

[6] D. X. 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] Y.-C. Chang and M. Mitzenmacher, “Privacy preserving keyword searches on remote encrypted data,” in Proceedings of the Third international conference on Applied Cryptography and Network Security. Springer-Verlag, 2005, pp. 442–455.

[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] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, “Fuzzy keyword search over encrypted data in cloud computing,” in INFOCOM, 2010 Proceedings IEEE. IEEE, 2010, pp. 1–5.

[10] M. Kuzu, M. S. Islam, and M. Kantarcioglu, “Efficient similarity search over encrypted data,” in Data Engineering (ICDE), 2012 IEEE 28th International Conference on. IEEE, 2012, pp. 1156–1167.

[11] C. Wang, K. Ren, S. Yu, and K. M. R. Urs, “Achieving usable and privacy-assured similarity search over outsourced cloud data,” in INFOCOM, 2012 Proceedings IEEE. IEEE, 2012, pp. 451–459.