TPA Scheme over Large Scale Cloud System for Achieving Secure and Robust Query Result and Session Verification

P. Fiona Swithin [1], M.Chella Rathi [2], S. Fathima Farveen [3], C.Manikandan [4]
[1, 2, 3] Student, Department of Computer Science and Engineering, University College of Engineering, Nagercoil, Tamil Nadu, India.
[4] Teaching Fellow, Department of Computer Science and Engineering, University College of Engineering, Nagercoil, Tamil Nadu, India.

Abstract – Secure search techniques over large scale cloud data allow an authorized user to query data files of interest by submitting encrypted query keywords to the cloud server in a privacy preserving manner. However, in practice, the returned query results may be incorrect or incomplete in the evasive cloud environment. Thus, a well-functioning secure semantic query system should provide a query results verification mechanism that allows the data user to verify results. In this project, proposed TPA scheme for secure design, easily adaptable, fine-grained query results with encrypted tags in semantic environment, sessions verification mechanism to maintain the particular session by generating bulk amount of keys, retrieving top k query generating bulk amount of keys, retrieving top k query results. The verification scheme is loose coupling to concrete secure semantic search techniques and can be very easily integrated into any secure query scheme. Furthermore, a multiple authority verification technique with extremely small storage cost is proposed to guarantee the authenticity and a verification key request technique is presented to allow the query user to securely obtain the desired verification key. Index terms: cloud computing, data search, encryption, session management, third party auditor.

1. INTRODUCTION

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources(e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. Driven by the abundant benefits brought by the cloud computing such as cost saving, quick deployment, flexible resource configuration, etc., more and more enterprises and individual users are taking into account migrating their private data and native applications to the cloud server (e.g.: [1]). A matter of public concern is how to guarantee the security of data that is outsourced to a remote cloud server and breaks away from the direct control of data owners. Encryption on private data before outsourcing is an effective measure to protect data confidentiality. However, encrypted data make effective data retrieval a very challenging task. To address the challenge (i.e., search on encrypted data), Song et al. first introduced the concept of searchable encryption and proposed a practical technique that allows users to search over encrypted data through encrypted query keywords (e.g.: [2], [3]). Later, many searchable encryption schemes were proposed based on symmetric key and public-key setting to strengthen security and improve query efficiency. Recently, with the growing popularity of cloud computing, how to securely and efficiently search over encrypted cloud data becomes a research focus. Some approaches have been proposed based on traditional searchable encryption schemes in which aim to protect data security and query privacies with better query efficient for cloud computing. However, all of these schemes are based on an ideal assumption that the cloud server is an" honest-but-curious" entity and keeps robust and secure software/hardware environments (e.g.: [6]). As a result, correct and complete query results always be unexceptionally returned from the cloud server when a query ends every time. However, in practical applications, the cloud server may return erroneous or incomplete query results once he behaves dishonestly for illegal profits such as saving computation and communication cost or due to possible software/hardware failure of the server. Therefore, the above fact usually motivates data users to verify the correctness and completeness of query results. Some researchers proposed to integrate the query results verification mechanisms to their secure search schemes (e.g., embedding verification information into the specified secure indexes or query results). Upon receiving query results, data users use specified verification information to verify their correctness and completeness. However, But, existing system has challenging task about effective data retrieval. On the other hand, it has less secure for data because existing system doesn't maintain session based log information and also the query result is incorrect and incomplete because the existing system doesn’t utilize proper algorithms. In a search process, for a returned query results set that contains multiple encrypted data files, a data user may wish to verify the correctness of each encrypted data file (thus, he can remove incorrect results and retain the correct ones as the ultima query results) or wants to check how many or which qualified data files are not returned on earth if the cloud server intentionally omits some query results. This information can be regarded as a hard evidence to punish the cloud server. This is challenging to achieve the fine-grained verifications since the query and verification are enforced in the encrypted environment. We proposed a secure and fine-grained query results verification scheme retrieving top k queries in semantic environment, maintaining session generating bulk amount of keys.

Our contributions

- We proposed Third Party Authority(TPA) scheme for secure design, easily adaptable, fine-grained query results with encrypted tags in semantic environment.
- Sessions verification mechanism to maintain the particular session by generating bulk amount of keys, retrieving top k query results.
- In case of any unauthorized access, notification containing the location is sent to the particular data user.
- The verification scheme is loose coupling to concrete secure semantic search techniques and can be very easily integrated into any secure query scheme.
- Multiple authority verification technique with extremely small storage cost is proposed to guarantee the authenticity. Verification key request technique is presented to allow the query user to securely obtain the desired verification key.

2. RELATED WORK

2.1 Secured Search in Cloud Computing

Essentially, the secure search is thus a technique that allows an authorized data user to search over the data owner's encrypted data by submitting encrypted query keywords in a privacy-preserving manner and is an effective extension of traditional searchable encryption i.e. semantic searchable encryption to adapt for the cloud computing environment (e.g.: [2], [3], [4], [5]). Motivated by the effective information retrieve on encrypted outsourced cloud data, Wang et al. first proposed a keyword-based secure search scheme and later the secure keyword search issues in cloud computing have been adequately researched which aim to continually improve search efficiency, reduce communication and computation cost, and enrich the category of search function with better security and privacy protection (e.g.: [6], [7]). A common basic assumption of all these schemes is that the cloud is considered to be an" honest-but-curious" entity as well as always keeps robust and secure software/hardware. As a result, under the ideal assumption, the correct and complete query results always be unexceptionally returned from the cloud server when a query ends every time. Here encryption and decryption are done on the basis of Cipher text policy Advanced Attribute based Encryption scheme to retrieve the requested data in a secured manner.

2.2 Verifiable secure search in cloud computing

In practical applications, the cloud server may return erroneous or false search results once he behaves dishonestly for illegal profits or due to possible software/hardware failure of the cloud server. Because of the possible data corruption under a dishonest setting, serval research works have been proposed to allow the data user to enforce query results verification in the secure search fields for cloud computing. So we use Third Party Auditor to verify both data owner and data user ignored to check the authenticity of both of them. Then TPA gets all details about the particular data user from Data owner to verify whether he/she is an authenticated user. In case of an authenticated user, TPA gives permission to the Cloud Service Provider to provide the requested data in encrypted format and also maintaining each and every session using session management scheme by Message Digest(MD-5) algorithm.

3. BACKGROUND

To clarify our proposed problems, in this section, we present our system model, threat model, and several preliminaries used to implement our scheme.

3.1 System Model

The system model of the secure search over encrypted cloud data usually includes four entities: data owners, data users, TPA and the cloud server, which describes the following scenario: data owners encrypt their private data and upload them to cloud server for enjoying the abundant benefits brought by the cloud computing as well as guaranteeing data security (e.g.: [8]). Meanwhile, the secure searchable indexes are also constructed to support effective keyword search over encrypted outsourced data. An authorized data user obtains interested data files from the cloud server by submitting query trapdoors (encrypted query keywords) to the cloud server, who performs search over secure indexes according to trapdoors and sends the query results to the data user. Third Party Auditor, verifies both data owner and data user in order to preserve authenticity.

Fig.1 explains, the overall architecture containing; first data owner doesn't handle all the requests from the data user. So we newly introduce Third Party Auditor (TPA). TPA initially verifies both data owner and data user whether they are authenticated or not. Data owner upload the files in encrypted formatted. The files such as videos, images, PDF and etc., are presented in encrypted tags format. When data user give request to get the desired files to data owner. Then the data owner forwards the request to the TPA.TPA verifies the details about the data user and give permission to Cloud Service Provider(CSP)to provide bulk amount of keys. The bulk amount of keys is generated based on every user's attributes and identity. In every session the data user uses anyone of that keys to login the system to request data owner to access the data. Then the data owner forwards the request to TPA then TPA verifies the key and again give permission to CSP to provide the requested data to data owner in encrypted format. Then data user use secret key to decrypt the data. Then the encrypted data becomes viewable after decryption.

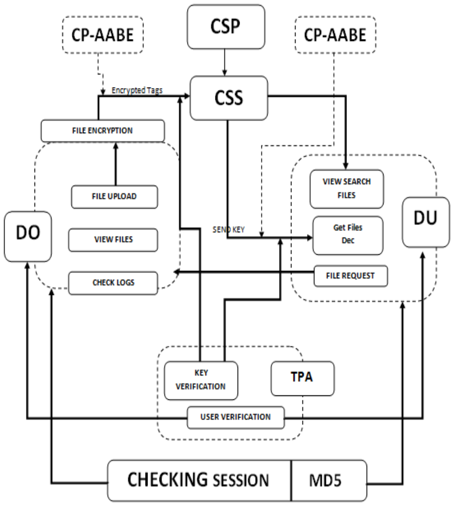


Fig 1. Overall architecture for secure search

DU =Data Use, DO=Data Owner, CS=Cloud Server, TPA=Third Party Auditor.

In this process in case of any attack the notification containing the location of the hacker is send to particular user's email.

3.2 Threat model

In this paper, compared with the previous works, an important distinction about the threat model is that the cloud is considered to be an untrusted entity. More specifically, first of all, the cloud server tries to gain some valuable information from encrypted data files, secure indexes, session management and verification scheme. Then, the cloud server would intentionally return false search results for saving computation resource or communication cost (e.g.: [10]). Since multiple data owners are used to handle every requests, if one data owner is busy other data owner will handle the requests and TPA produce bulk amount of keys, hacker is not able to predict what key user uses at a particular session so forgery will not happen and if sometimes hacker predicts the key the data will be in encrypted format only so security is ensured and also the notification is sent to user.

4. PSEUDOCODE

4.1 Encryption

```
Set index I = 0 for formed expression
For (length of formed expression)
Switch(expression(i))
Case(i)
Expression(i)==numeric data
Expression(i)==(A or B or C or D or E or F or G or H or I)
Case(ii)
Expression(i)==arithmetic operator
Expression(i)=(K or M or Q or P)
Case (iii)
Expression(i) = Power
Expression(i) = R
Case(iv)
Expression(i)==floating point
Expression(i)=w
End switch statement
End for statement
Return expression
```

4.2 Decryption

```
Set index i=0 for encrypted data sequences
For length of encrypted data sequences
Switch(expression(i))
Case(i)
Expression(i)==(A or B or C or D or E or F or G or H or I)
Expression(i)=(0 to 9)
Case(ii)
Expression(i)==(K or M or Q or P)
Expression(i)=arithmetic operator
Case(iii)
Expression(i)==R
Expression(i)==Power
Case(iv)
Expression(i)=W
Expression(i)=floating point
End switch statement
End for statement
Return plain text
```

5. ALGORITHM

5.1 CP-Advanced Attribute Based Encryption

Advanced Attribute based encryption is an important security concept that can be applied to almost any role based system today to provide data confidentiality and integrity. The concept of attribute-based encryption was first proposed in a landmark work by Amit Sanai and Brent Waters and later by Vipul Goyal, Omkant Pandey, Amit Sahai and Brent Waters. It is a type of public-key encryption in which the secret key of a user and the cipher text are dependent upon attributes (e.g. the country he lives, or the kind of subscription he has). In such a system, the decryption of a cipher text is possible only if the set of attributes of the user key matches the attributes of the cipher text. A crucial security feature of Attribute-Based Encryption is collusion-resistance: An adversary that holds multiple keys should only be able to access data if at least one individual key grants access.

In ABE policies are described through attributes (age, relationship, trust, location, etc.). From an OSN perspective, the categories of social connections are treated as attributes. Adam is able to encrypt the data in such a way that only if the attributes match a given key then the data can be decrypted. Users colluding cannot decrypt the data. ABE uses a tree-based access structure which must be satisfied with a given set of attributes in order to decrypt the data. The tree-based access structure allows the encryptor to specify which attributes can decrypt the data. It uses operators such as AND, OR and k-of-n. AND is usually known as' n of n' and OR is known as '1 of n'. For example, if Adam wants to encrypt a data P such that only someone with the attributes friend AND colleague or the attribute family can decrypt it, the tree-based access structure.

5.2 Search Steps

1) Enter Search
2) Keyword split into multiple phase
3) Keyword send to query processing engine
4) QPE filter the validating keyword
5) Compare with Large Scale Database
6) Send Response in XML http Format
7) Parsing Response
8) Display a Result

The query request is entered in the search tab which is in encrypted format so that is keyword is divided to multiple phases. Then the keyword is sent to query processing engine(QPE). The keyword is filtered by QPE which is compared with the larger scale database if its valid, the response in XML HTTP format. Then response is parsed and the result is displayed.

5.3 Performance Evaluation

The figures Fig 2 and Fig 3 represent the graphical representations of the comparison of performance evaluation of proposed and existing system. The existing system does not support highly scalable efficient search on large database. When document collection is too large, the collection will be divided into sub- collections and stored in different servers, which makes the ranking process to be delayed. It does not support Integrity check in rank order in the search result, when the cloud server is untrusted. It does not support secure search index. The search time depends on the number of documents in the data set.

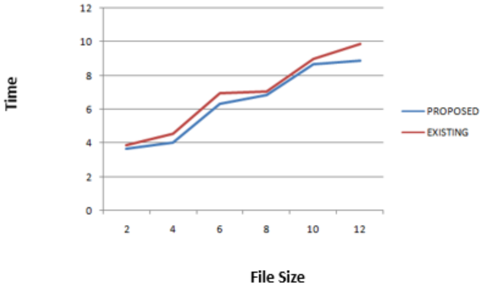
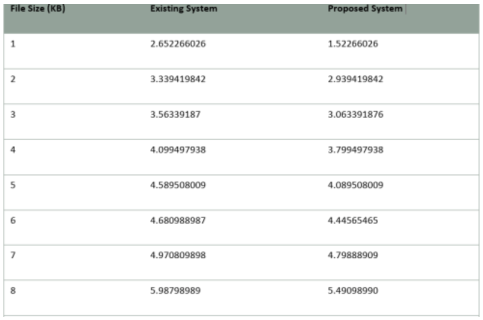


Fig.2: Graphical representation of File size vs UploadedTime.



File Size (KB)	Existing System	Proposed System
1	2.652266026	1.52266026
2	3.339419842	2.939419842
3	3.56339187	3.063391876
4	4.099497938	3.799497938
5	4.589508009	4.089508009
6	4.680988987	4.44565465
7	4.970809898	4.79888909
8	5.98798989	5.49098990

Table 1:Upload time of existing time vs proposed system with respect to file size

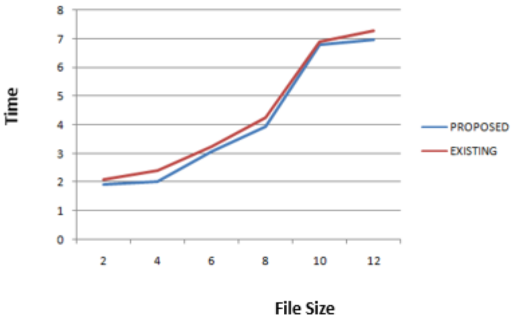
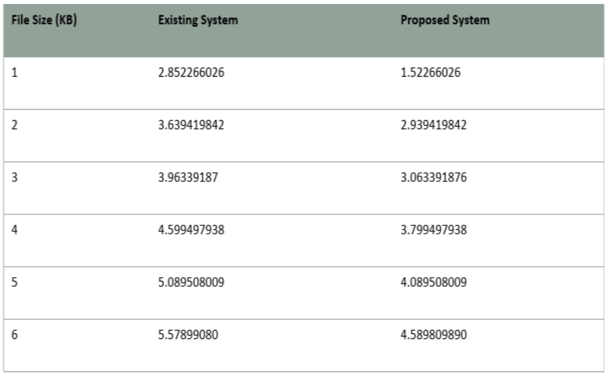


Fig.3: Graphical representation of File size vs Encryption Time.

CertP,a = d“P k a”.

C= (EPKP [κ1], E “P k a”[κ2]).



File Size (KB)	Existing System	Proposed System
1	2.852266026	1.52266026
2	3.639419842	2.939419842
3	3.96339187	3.063391876
4	4.599497938	3.799497938
5	5.089508009	4.089508009
6	5.57899080	4.589809890

Table2: Encryption time of existing time vs proposed system with respect to file size

This proposed system supports all environment like large scale distributed systems. Here, the data is not divided even though replicate to multiple server. If any server cannot able to response, the request will pass to other server. It supports Index based search. Every transaction, integrity will check. The search time depends on the query set not data set.

6. CONCLUSION

In this paper, we propose a secure, easily integrated, and fine-grained TPA scheme for secure search over encrypted cloud data with session verification for secure authorization. Different from previous works, we design a session verification object request technique, by which the cloud server knows nothing about which user is requested by the data user and actually returned by the cloud server. Performance and accuracy experiments demonstrate the validity and efficiency of our proposed scheme.

REFERENCES

[1] P. Mell and T. Grance, “The nist definition of cloud computing,” http://dx.doi.org/10.602/NIST.SP.800-145.

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

[3] S. Kamara and K. Lauter, “Cryptographic cloud storage,” in Springer RLCPS, January 2010.

[4] D. Song, D. Wagner, and A. Perrig, “Practical techniques for searches on encrypted data,” in IEEE Symposiumon Security and Privacy, vol. 8, 2000, pp. 44–55.

[5] E.-J. Goh, “Secure indexes,” IACR ePrint Cryptography Archive, http://eprint.iacr.org/2003/216, Tech. Rep., 2003

[6] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, “Public-key encryption with keyword search,” in EUROCRYPR, 2004, pp. 506–522.

[7] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, “Searchable symmetric encryption: improved deinitions and efficient constructions,” in ACM CCS, vol. 19, 2006, pp. 79–88.

[8] M. Bellare, A. Boldyreva, and A. O’Neill, “Deterministic and efficiently searchable encryption,” in Springer CRYPTO, 2007.

[9] K. Kurosawa and Y. Ohtaki, “Uc-secure searchable symmetric encryption,” Lecture Notes in Computer Science, vol. 7397, pp. 258–274, 2012.

[10] P. Xu, H. Jin, Q. Wu, and W. Wang, “Public-key encryption with fuzzy keyword search: A provably secure scheme under keyword guessing attack,” IEEE Transactions on Computers, vol. 62, no. 11,pp. 2266–2277, 2013.