International Journal of
Scientific Engineering and Technology Research

ISSN 2319-8885
Vol.07,Issue.04,
April-2018,
Pages:0715-0718

www.ijsetr.com

Publicly Verifiable Boolean Query over Outsourced Encrypted Data

B. SHILPA[1], A. MALLIKARJUNA[2], S. RAMAKRISHNA[3]

[1]PG Scholar, Dept of Computer Science, S.V.University, Tirupati, AP, India.
[2]Teaching Assistant, Dept of Computer Science, S.V.University, Tirupati, AP, India.
[3] Professor, Dept of Computer Science, S.V.University, Tirupati, AP, India.

Abstract: Outsourcing storage and computation to the cloud has become a common practice for businesses and individuals. As the cloud is semi-trusted or susceptible to attacks, many researches suggest that the outsourced data should be encrypted and then retrieved by using searchable symmetric encryption (SSE) schemes. Since the cloud is not fully trusted, we doubt whether it would always process queries correctly or not. Therefore, there is a need for users to verify their query results. Motivated by this, in this paper, we propose a publicly verifiable dynamic searchable symmetric encryption scheme based on the accumulation tree. We first construct an accumulation tree based on encrypted data and then outsource both of them to the cloud. Next, during the search operation, the cloud generates the corresponding proof according to the query result by mapping Boolean query operations to set operations, while keeping privacy-preservation and achieving the verification requirements: freshness, authenticity, and completeness. Finally, we extend our scheme by dividing the accumulation tree into different small accumulation trees to make our scheme scalable. The security analysis and performance evaluation show that the proposed scheme is secure and practical.

Keywords: Cloud Computing, Outsourced Encrypted Data, Query Integrity Verification.

I. INTRODUCTION

The great flexibility and economic savings of cloud computing motivate companies and individuals to outsource their data to cloud servers. By outsourcing a dataset to the cloud, the data owner or other valid data users can then issue the cloud informational queries that are answered according to the dataset. This model captures a variety of real-world applications such as outsourced SQL queries, streaming dataset, and outsourced file systems. However, the privacy and confidentiality concerns of data often make them reluctant to do that. Therefore, it is natural for a data owner to encrypt data before outsourcing them to the cloud. As a result, the cloud server cannot reveal the content of the outsourced data. However, since data has been encrypted before outsourced the cloud, it obstructs the traditional data utilization service based on plaintext keyword search. Thus, how to efficiently obtain encrypted data from the cloud server is very important for outsourcing storage applications. Motivated by this challenge, many searchable symmetric encryption (SSE) schemes are proposed to satisfy different search functions on encrypted data. By using these schemes, a data owner can outsource encrypted data to protect the confidentiality of data, and data users can query from the cloud. Specifically, a pharmaceutical company would like to outsource a set of sanitized records of its clinical trials to the cloud server after encryption. Then, its employees or external third parties such as the European Medicines Agency use these information for research or other operations by keywords search.

However, due to the nature of the delegation/outsourcing, the cloud server can fully control the outsourced data and decide the SSE query result for the company or other third parties, which causes issues of trust. There are several reasons for which the data owner/users cannot trust the cloud server: Firstly, the cloud server may run buggy software or its systems may be vulnerable to security breaches, leading to incorrect result. Secondly, in some applications, the company or third parties want to rule out accidental errors during the computation. Finally, when there is a legal dispute between the pharmaceutical company and a patient, the pharmaceutical company may collude with the cloud to provide an incomplete search result to the data user (such as a judge). Therefore, it is necessary for the cloud server to have the ability to provide proofs with which the data owner/users are able to verify the integrity of the search result returned by the cloud server. As mentioned above, the query integrity is significantly important to the data owner/users. To achieve this purpose, we allow the data owner to perform some polynomial-time preprocessing on encrypted data before outsourcing them to the cloud and to save a small verification state that allows data users to verify the returned proof provided by the cloud server. When issuing an update operation, the data owner will update its verification state at the same time. If the verification state can be made by any third-party (not necessarily the user originating the search query), we say that the proof is publicly verifiable. Public verifiability is particularly important in multi-user settings.

The query integrity verification has been studied for structured attributed-value type database and streaming setting. Many verification schemes are proposed to meet verification requirements (details are introduced in related work). However, these schemes may not be suitable for SSE condition:

- These schemes assumed that the data stored at the third party publisher is in plaintext, while for SSE, the data is encrypted so that the publisher cannot "see" the actual content;
- All of them are ordered by some sequences or the data update according to time slices which makes verification easier.

Obviously, the query integrity verification in SSE should contain three aspects:

- **Freshness:** The record values in the answer must be up-to-date;
- **Authenticity:** Every record returned in the search result must originate from the data owner;
- **Completeness:** Every record that satisfies the query condition must be in the search result.

Moreover, the verification process should be secure and privacy-preserving. Even though there are many studies about the SSE, the query integrity verification work is limited except. Moving a step forward, in this paper, we present a publicly verifiable scheme which can publicly verify whether the cloud server has faithfully executed Boolean search operations in dynamic SSE (DSSE). To realize query integrity verification in DSSE, a possible solution is that we can map query operations to set operations, so that the query integrity verification can be achieved by using the accumulation tree. However, there are several challenges that must be overcome before we use this method: Firstly, how to map query operations to set operations in DSSE. Secondly, how to construct the accumulation tree with limited information of DSSE so that the construction process would not affect the architecture of DSSE. Thirdly, how to achieve the secure and privacy-preserving query integrity verification by using the accumulation tree in DSSE while ensuring the verification requirements. Finally, how to ensure the efficiency and scalability of the query integrity verification process in DSSE. All these challenges make the query integrity verification of DSSE different from the existing work. Thus, the contributions of our work are listed as follows:

- We propose a publicly Boolean query integrity verification scheme over the outsourced dynamic encrypted data to check the freshness, authenticity, and completeness of the query result.
- We construct an accumulation tree, a special Merkle hash tree, in DSSE to map Boolean operations of keywords to set operations, which is different from the traditional signature or aggregate signature.
- We extend our scheme by dividing encrypted data into small groups, then constructing the corresponding accumulation trees for each group to make our scheme efficient and scalable solutions.
- The security and performance analysis are carried out to show that the proposed scheme is privacy-preserving and practical.

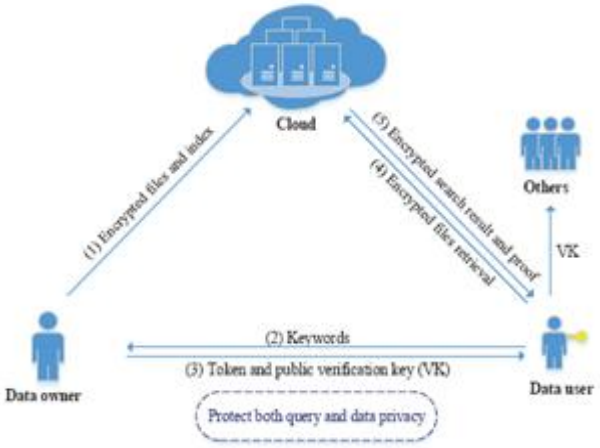

Fig.1. The System Model.

II. SYSTEM MODEL AND PRELIMINARIES

A. System Model and Motivation

A database DB=(ind_i, W_i) is a list of identifier and keyword-set pairs, where ind_i is a document identifier and W_i is a list of keywords in that document. The system model is illustrated as Fig. 1 which contains three entities: a data owner DO, who outsources a large-scale collection of e d documents to the remote cloud server S; the cloud server S, which provides storage services; and data users DUs, who can enjoy the documents from the cloud. Considering the privacy problem and the efficient information retrieval, the cloud-based data storage and sharing process are described as follows:

- The data owner DO first extracts the keywords of each document and builds a keyword index. Then DO encrypts the documents as well as the keyword index. Notice that the documents are dynamic. That is, at any time the data owner can add, modify or delete one or more documents from the cloud.
- After the data owner outsources the encrypted documents as well as the encrypted keyword index to the cloud server, the data users DUs can use the Boolean query expression to query and retrieve documents of interest from the cloud. To this end, the data users should first pass the authentication of the data owner, and then get an encrypted search token according to the Boolean query expression and the public verification key (V K).
- After receiving the query token, the cloud server S executes the query and returns the encrypted documents according to the token. Moreover, to verify whether the cloud has correctly executed the search operation or not, an additional proof is also appended to the result.
- When receiving the result and the corresponding proof, the data users DUs or others can verify the correctness of the search result by V K. Finally, DUs can decrypt encrypted documents after the verification is correct.

B. Attack Model

The data owners are naturally trusted. Both authorized and unauthorized data users are semi-trusted, meaning that they may try to infer some sensitive information from the query result and the corresponding proof. The cloud server is not trusted as it executes the search operations, which already implies that the cloud may manipulate the outsourced encrypted data. Moreover, we also consider potential malicious data users which could collude with the cloud server or other malicious users, or help the cloud to cheat with other users. Note that the assumption about malicious data users enables the public verifiability property of our solution. Obviously, our attack model is more general.

C. Design Objective

The objective of our work is to design a publicly Boolean query integrity verification scheme in DSSE. First, we present a leakage function L, which covers all the information leakage in our scheme. For instance, the privacy leakage introduced by a Boolean query Q on DB is denoted as L(DB,Q). Informally, the privacy leakage function for the query integrity verification in DSSE includes the following aspects:

- **Size Pattern:** The cloud server can learn the total number of data records in the database and the total number of search queries submitted by each data user;
- **Access Pattern:** The cloud server reveals the identifier of each encrypted data record that is returned for each query ;
- **Search Pattern:** The cloud server can learn if the same encrypted data record is retrieved by two different queries;
- **Path Pattern:** The cloud server learns the set of nodes of the accumulation tree in each query integrity verification.

Notice that the size pattern, access pattern, and search pattern are general information leakage in searchable encryption. Path pattern is introduced for generating the proof for query integrity verification. We argue that revealing path pattern is only a minor leakage in the privacy-preserving Boolean query since the information we use to generate the proof is in the cloud server, which has little influence on the data privacy. To sum up, the design objective is to achieve secure and privacy-preserving Boolean query integrity verification in DSSE under the leakage function L.

III. PUBLICLY VERIFIABLE BOOLEAN QUERY OVER ENCRYPTED DATA

We define and construct our publicly verifiable scheme in DSSE. In order to introduce our scheme, we first describe the construction process of the query integrity verification with the SSE scheme, i.e., OSPIR-OXT.

A. Definition

Based on OSPIR-OXT, a publicly verifiable DSSE contains an algorithm EDBSetup (DB, RDK), an algorithm Update (EDB, AT), a protocol GenToken(K,w), a protocol Search(token), and an algorithm Verify(a(q), Π). The syntax is described as follows:

- **(K, s, EDB, AT) ← EDBSetup(DB, RDK):** This algorithm can be divided into two phases: First, it takes a security parameter λ as input and outputs a secret key K and a secret key s for the accumulation tree construction, Second, it takes a database DB as input, and outputs an encrypted database EDB and a constructed accumulation tree AT. Then both EDB and AT are stored at the cloud server.
- **(EDB′, AT′) ← Update(EDB, AT):** This algorithm runs between the data owner and the cloud server. The data owner inputs an update operation, the update index ind_i of the ith document, and the list of unique keywords in the document. The protocol adds or deletes the document from the EDB and updates the corresponding AT.
- **(token, V Kw) ← GenToken(K,w):** This protocol is executed between the data owner and the data user. For an authorized user with the Boolean query $w = w_1, \cdots, w_n$, the data owner uses K to generate the corresponding token to enable the search at the cloud server for the data user and the corresponding public verification key V Kw.
- **(Π, a(q)) ← Search(token):** This protocol is run by the cloud server to conduct the search operation over encrypted index according to the token of the data user. The server returns the search result a(q) and the proof Π. Notice that the proof can be an optional item.
- **(0, 1) ← Verify(a(q),Π, V Kw):** This algorithm is run by the verifier to verify whether the server has faithfully executed the search operations or not according to the search token by the proof and V Kw.

B. Overview

The basic idea underlying the construction is that: to achieve the public Boolean query integrity verification for outsourced encryption documents, an accumulation tree is associated with encrypted documents. Then the data owner outsources both of them to the cloud server. During the search process, after executing the privacy-preserving Boolean query over encrypted data, the cloud should compute its corresponding proof according to the query result by using the accumulation tree. To compute the proof, the cloud maps relations of Boolean queries to the similar set operations of the accumulation tree. Finally, with the search result, the proof and the public verification key, the data user or others can verify the freshness, authenticity, and completeness of the search result even without decrypting them.

IV. RELATED WORK

We introduce related work in three aspects: searchable encryption, query integrity verification, and query integrity verification on encrypted data. Searchable encryption: Song explicitly considered the problem of searchable encryption and presented a scheme with search time that was linear with the size of the data collection for the first time. Their construction supports insertions/deletions of documents in a straightforward way. Curtmola gave the first index-based SSE constructions to achieve sublinear search time for SSE. A similar construction was proposed by Chase and Kamara,

but with higher space complexity. Subsequently, Kamara introduced a dynamic scheme which was the first one with sublinear search time. However, it did not achieve forward privacy or revealed hashes of the unique keywords contained in the document during the update. Recently, Cash presented a SSE scheme supporting conjunction queries over static data. Based on Cash work, Jarecki proposed a scheme that allowed data owners to authorize third parties and to execute private information retrieval on the outsourced database. Faber extended the search capabilities of the system from by supporting range queries, substring queries, wildcard queries, and so on. Moreover, they also extended their techniques to the more involved multi-client SSE scenarios studied. However, they did not consider the update process and the integrity query verification comparing with our scheme. To support efficient update and preserve the privacy during the update process. It is obvious that, all the proposed schemes for SSE do not consider the verification problem of the search result.

Query Integrity Verification: Li introduced an efficient implementation of Merkle hash tree authenticated B+-tree to audit the completeness of the query result, and demonstrated its superiority over signature chaining. By using signature aggregation, Pang proposed a scheme to verify the freshness, authenticity, and completeness of query answers from frequently updated databases that were hosted on untrusted servers. Azraoui used the well-established techniques of Cuckoo hashing, polynomial-based accumulators, and Merkle trees to publicly verify conjunctive keyword search in outsourced databases. However, unlike our solution, none of these solutions achieve public verification for encrypted data.

V. CONCLUSION

We study the problem of verifying the freshness, authenticity, and completeness of the Boolean query result over the outsourced encrypted data. Based on OSPIROXT, we propose a publicly verifiable scheme by constructing the accumulation tree to achieve the query integrity verification while keeping privacy-preserving and efficiently practical. The security analysis shows that without protecting the access pattern, our scheme can keep the privacy-preserving of private information retrieval. The performance demonstrates our scheme is scalable.

VI. REFERENCES

[1] S. Jiang, X. Zhu, L. Guo, and J. Liu, "Publicly verifiable boolean query over outsourced encrypted data," in Proceedings of the 2015 IEEE Global Communications Conference (GLOBECOM 2015). IEEE, 2015.

[2] S. Jarecki, C. Jutla, H. Krawczyk, M. Rosu, and M. Steiner, "Outsourced symmetric private information retrieval," in Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. ACM, 2013, pp. 875–888.

[3] S. Kamara, C. Papamanthou, and T. Roeder, "Dynamic searchable symmetric encryption," in Proceedings of the 2012 ACM conference on Computer and communications security. ACM, 2012, pp. 965–976.

[4] D. Cash, S. Jarecki, C. Jutla, H. Krawczyk, M. Rosu, and M. Steiner, "Highly-scalable searchable symmetric encryption with support for boolean queries," in Advances in Cryptology- CRYPTO 2013. Springer, 2013, pp. 353–373.

[5] M. Naveed, M. Prabhakaran, and C. A. Gunter, "Dynamic searchable encryption via blind storage." in Proceedings of the IEEE Symposium on Security and Privacy, San Jose, CA, May, 2014.

[6] E. Stefanov, C. Papamanthou, and E. Shi, "Practical dynamic searchable encryption with small leakage." IACR Cryptology ePrint Archive, vol. 2013, p. 832, 2013.

[7] D. Cash, J. Jaeger, S. Jarecki, C. S. Jutla, H. Krawczyk, M. C. Rosu, and M. Steiner, "Dynamic searchable encryption in verylarge databases: Data structures and implementation," in 21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, California, USA, February 23-26, 2014.

[8] S. Faber, S. Jarecki, H. Krawczyk, Q. Nguyen, M. Rosu, and M. Steiner, "Rich queries on encrypted data: Beyond exact matches," in European Symposium on Research in Computer Security (ESORICS 2015). Springer, 2015, pp. 123–145.

[9] M. Azraoui, K. Elkhiyaoui, M. Onen, and R. Molva, "Publicly verifiable conjunctive keyword search in outsourced databases," in Proceedings of the 2015 IEEE conference on Communications and Network Security (CNS). IEEE, 2015, pp. 619–627.

[10] http://www.wired.com/2009/01/magnolia-suffer/.

[11] http://mashable.com/2011/02/27/gmail-glitch/.

[12] B. Parno, M. Raykova, and V. Vaikuntanathan, "How to delegate and verify in public: Verifiable computation from attribute-based encryption," in Proceedings of the 9th Theory of Cryptography Conference, TCC 12, 2012, pp. 422–439.

[13] H. Pang, j. Zhang, and k. Mouratidis, "Scalable verification for outsourced dynamic databases," Proceedings of the VLDB Endowment, vol. 2, no. 1, pp. 802–813, 2009.

Author's Profile:

B. Shilpa, MCA Student, Dept of Computer Science, Sri Venkateswara University, Tirupati, Andhrapradesh, India.
Email: bsilpa506@gmail.com

A.Mallikarjuna, Teaching Assistant Department of Computer Science, Sri Venkateswara University College of Commerce Management and Computer Science, S.V University, Tirupati, AP, INDIA. Email: mallisvu9@gmail.com

Dr.S.Ramakrishna M.Sc, M.Phil, M.Tech, Ph.D., Working as a professor in Department of Computer Science, Sri Venkateswara University College of Commerce, Management and Computer Science, Tirupati (AP), INDIA.
Email: drsrsmskrishna@yahoo.com.