2021 8th International Conference on Future Internet of Things and Cloud (FiCloud) | 978-1-6654-2574-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/FICLOUD49777.2021.00023

Access Pattern Hiding in Searchable Encryption

Fateh Boucenna
CERIST, research center, Algeria
Email: fboucenna@cerist.dz

Omar Nouali
CERIST, research center, Algeria
Email: onouali@cerist.dz

Kamel Adi
UQO, Qc, Canada
Email: kamel.adi@uqo.ca

Samir Kechid
USTHB university, Algeria
Email: skechid@usthb.dz

***Abstract*—Cloud computing is a technology that provides users with a large storage space and an enormous computing power. For privacy purpose, the sensitive data should be encrypted before being outsourced to the cloud. To search over the outsourced data, *searchable encryption* (SE) schemes have been proposed in the literature. An SE scheme should perform searches over encrypted data without causing any sensitive information leakage. To this end, a few security constraints were elaborated to guarantee the security of the SE schemes, namely, the keyword privacy, the trapdoor unlinkability, and the *access pattern*. The latter is very hard to be respected and most approaches fail to guarantee the access pattern constraint when performing a search. This constraint consists in hiding from the server the search result returned to the user. The non respect of this constraint may cause sensitive information leakage as demonstrated in the literature. To fix this security lack, we propose a method that allows to securely request and receive the needed documents from the server after performing a search. The proposed method that we call the *access pattern hiding* (APH) technique allows to respect the access pattern constraint. An experimental study is conducted to validate the APH technique.**

I. Introduction

Nowadays, companies and individuals demand an increasing amount of storage space for their data and computing power for their applications. For this purpose, several new technologies have been designed and implemented such as the cloud computing. The latter provides its users with a storage space and computing power according to their needs in a flexible and personalized way. However, the outsourced data such as emails, electronic health records, and company reports are sensitive and confidential, hence, they must be encrypted before being sent to the cloud.

To perform a search over these data, it is no longer possible to exploit traditional search engines, given that the outsourced data are encrypted. Consequently, lots of searchable encryption (SE) schemes have been proposed in the literature [1], [2], [3], [4], [5]. The common point between these approaches is that the *data owner* starts by encrypting the data collection and the generated index before outsourcing them. The *cloud server* exploits the encrypted index after receiving a trapdoor (encrypted query) in order to retrieve and return a top-k document identifiers to the *data user*. Finally, the latter asks the server to return him the needed documents among the top-k ones. In addition, the search process should be performed without decrypting any data and without causing any sensitive information leakage.

Furthermore, when designing an SE scheme, it is important to take into consideration some security constraints [1], [2]. The first one is called the keyword privacy and consists in preventing the server from making any link between terms and documents. The second one is called trapdoor unlinkability and consists in preventing the server from making links between a set of trapdoors. Finally, the *access pattern* constraint consists in hiding the search results from the server. Both the keyword privacy and the trapdoor unlinkability constraints are respected by most recent approaches. Nevertheless, it is very difficult to design an SE scheme that hides the access pattern from the server.

Some approaches tried to guarantee the access pattern constraint by encrypting the *search result* [6], [7]. However, after receiving and decrypting the result, the authorized user might request some relevant documents from the server. Consequently, a part of the search result (the requested documents set) is revealed to the cloud and thus, the access pattern constraint is not fully respected even if the search result is encrypted. Unfortunately, the non-respect of the access pattern constraint may cause sensitive information leakage as demonstrated by Islam et al. [8] in their proposed attack.

To solve this problem, a few techniques have been proposed in the literature. One of these techniques is called the *blind storage* [9] which consists in splitting each document into several blocks in order to avoid the server from knowing the number of documents or distinguishing between them. However, the drawback of this technique is that a document is recognized by the server as soon as it is accessed by a user. Consequently, this technique does not guarantee the access pattern constraint. Another approach proposed by Kellaris et al. [10] allows preserving privacy while accessing data by combining two techniques which are the differential privacy [11] and the oblivious RAM [12]. Nevertheless, the oblivious RAM dramatically decreases the search performance which makes this approach impracticable.

The aim of this work is to secure the access to the outsourced data after performing a search in order to guarantee a full respect of the access pattern constraint. Our method is applicable in any approach that respects the access pattern constraint *during* the search and needs a secure access to the data collection. Our idea consists in breaking any link on the cloud side between the index and the data collection as well as between the returned result and the accessed documents[1]. This new method prevents the server from deducing the search

[1] The set of documents which are requested by the user after performing a search.

978-1-6654-2574-2/21/$31.00 ©2021 IEEE
DOI 10.1109/FiCloud49777.2021.00023