Future Generation Computer Systems 72 (2017) 208–218

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

A multi-user searchable encryption scheme with keyword authorization in a cloud storage

Zuojie Deng [a,b,*], Kenli Li [c,d], Keqin Li [c,d,e], Jingli Zhou [a]

[a] School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China
[b] School of Computer and Communication, Hunan Institute of Engineering, Xiangtan, Hunan, 411104, China
[c] College of Information Science and Engineering, Hunan University, Changsha, Hunan 410082, China
[d] National Supercomputing Center in Changsha, Changsha, Hunan 410082, China
[e] Department of Computer Science, State University of New York, New Paltz, NY 12561, United States

HIGHLIGHTS

- A security model of keyword authorization search over encrypted files is defined.
- We propose a multi-user searchable encryption scheme with keyword authorization.
- To describe keyword authorization relationships, a KABtree is defined.
- We construct MSESK with asymmetric bilinear map groups of Type-3 and KABtrees.
- The performance evaluation experiments explain the feasibility of MSESKA.

ARTICLE INFO

Article history:
Received 28 October 2015
Received in revised form
7 March 2016
Accepted 18 May 2016
Available online 1 June 2016

Keywords:
Cloud storage
Encrypted data
Keyword authorization
Multi-user searchable encryption

ABSTRACT

Multi-user searchable encryption (MSE) allows a user to encrypt its files in such a way that these files can be searched by other users that have been authorized by the user. The most immediate application of MSE is to cloud storage, where it enables a user to securely outsource its files to an untrusted cloud storage provider without sacrificing the ability to share and search over it. Any practical MSE scheme should satisfy the following properties: concise indexes, sublinear search time, security of data hiding and trapdoor hiding, and the ability to efficiently authorize or revoke a user to search over a file. Unfortunately, there exists no MSE scheme to achieve all these properties at the same time. This seriously affects the practical value of MSE and prevents it from deploying in a concrete cloud storage system. To resolve this problem, we propose the first MSE scheme to satisfy all the properties outlined above. Our scheme can enable a user to authorize other users to search for a subset of keywords in encrypted form. We use asymmetric bilinear map groups of Type-3 and keyword authorization binary tree (KABtree) to construct this scheme that achieves better performance. We implement our scheme and conduct performance evaluation, demonstrating that our scheme is very efficient and ready to be deployed.

1. Introduction

1.1. Motivation

Cloud storage has become a prevalent storage scheme in recent years, where a user can store and share its files [1]. However, the cloud storage provider is not fully trusted, if a user outsources its confidential files to a remote cloud storage server in plaintext form, it may cause some horrible privacy leakage. A promising approach to protect confidential files for a user in a cloud storage is to encrypt its files by using a secure symmetric encryption algorithm, e.g. AES. However, storing files in encrypted form will make some useful file operation functions, such as search, sharing, etc., unavailable. If a user cannot share and search over its files on a remote cloud storage server, it will be reluctant to outsource its files to the cloud storage.

To resolve the searchable problem of encrypted files in a cloud storage server, we can use the searchable encryption technology

* Corresponding author at: School of Computer of Science and Technology, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China.
E-mail addresses: zjdeng@hotmail.com (Z. Deng), lkl@hnu.edu.cn (K. Li), lik@newpaltz.edu (K. Li), jlzhou@hust.edu.cn (J. Zhou).

http://dx.doi.org/10.1016/j.future.2016.05.017
0167-739X/

in the literature [2–9]. A user can store its files in encrypted form at an untrusted cloud server by using these searchable encryption schemes, and delegate the cloud server to search over its files by issuing a trapdoor (i.e. encrypted keyword). However, all these schemes are limited to the single-user setting where the file owner who generates these encrypted files is also the single-user performs searches on it. Therefore, their schemes cannot support encrypted file sharing. Generally speaking, there exists file sharing between a set of users in the cloud storage, and the file owner should authorize other users to search over its encrypted files. Since above schemes do not support this, we cannot apply these schemes to cloud storage directly. Curtmola et al. suggested to share the secure key for file search among some users by extending their single-user scheme directly [4]. Later, other researchers proposed some schemes for multi-user [10–15]. In all these schemes, there exists a user manager to manage the search capabilities of multiple users (e.g. enable them to search each other's files). However, there usually exists no trusted user administrator in a cloud storage, so all these multi-user schemes cannot directly be applied to cloud storage setting as well. To resolve this problem, Popa et al. proposed a searchable encryption scheme that enables keyword search on files encrypted with different keys [16]. But the granularity of authorization in their scheme is very coarse, and they did not explicitly specify how u_i can authorize u_j to search its indexes, where $i \neq j$. Subsequently, Tang extended the Popa–Zeldovich scheme, and proposed a secure and scalable multi-party searchable encryption scheme [17]. However, there are still three shortcomings in the scheme as follows:

- Firstly, since the authorization in the scheme is granted on the index level, the authorization granularity is also coarse. If u_i wants to authorize u_j to search for a subset of keywords in its indexes, then the scheme cannot complete this task.
- Secondly, the scheme only supports search authorization, but it does not support search authorization revocation explicitly.
- Thirdly, the match algorithm in the construction of the scheme has two pairing map operations, which seriously affect its performance.

1.2. Our contributions

In this work, we study the problem of how to enable a user to share its files with others and authorize them to search its files using a subset of keywords in encrypted form. We propose a multi-user searchable encryption scheme with keyword authorization in a cloud storage (MSESKA), which can be regarded as multi-user version of the symmetric searchable encryption proposed by Song et al. [2]. Briefly, our MSESKA allows every user to build an encrypted index for each of its files and store it on a cloud storage server. The index contains a list of encrypted keywords which are well organized, and some authorization information selectively authorizes other users to search for a subset of keywords in the index. Our contribution can be summarized as follows:

- Firstly, we define a formal security model of how to authorize a user to search for a word over an encrypted file in a cloud storage. In particular, our definition captures a strong noting of security, which is adaptive security against chosen-keyword attacks.
- Secondly, we propose a multi-user searchable encryption scheme with keyword authorization in a cloud storage, which supports keyword authorization revocation explicitly. Our scheme overcomes shortcomings in the Tang scheme [17].
- Thirdly, we propose a KABtree, and use it to organize the index in our scheme. If there exist n users and m keywords in a KABtree, then the construction time of the KABtree is $O(mn)$, the authorization or revocation time for a keyword of the KABtree is $O(m \log n)$, and the search time for a keyword of the KABtree is $O(r \log n)$, where r is the number of users that have been authorized to the keyword.
- Fourthly, we construct the MSESKA using asymmetric bilinear map groups of Type-3 [18] and prove that the construction is secure in the random oracle model under the BDHV and SXDH assumptions.
- Fifthly, we implement our scheme and conduct performance evaluation. The results show that our scheme is very efficient and practical.

The paper is organized as follows. In Section 2, we present the preliminary knowledge. In Section 3, we describe and give some definitions about the problem and define a multi-user searchable encryption scheme with keyword authorization in a cloud storage. The keyword authorization binary tree is presented in Section 4. In Section 5, we give a construction for MSESKA using Type-3 pairings and KABtree. In Section 6, we give the security and performance analysis for our construction. In Section 7, we implement our scheme and conduct a performance evaluation and performance evaluation results are presented here. In Section 8, we discuss related works. Finally, some conclusions are given in Section 9.

2. Preliminary

2.1. Notations

In this paper, we use the following notation. k is the security parameter. $p.p.t.$ denotes probabilistic polynomial time, $x \parallel y$ denotes the concatenation of x and y, and $y \xleftarrow{\$} A(x_1, x_2, \ldots, x_n; o_1, o_2, \ldots, o_n)$ denotes that y is the output of the algorithm A which runs with the input $x_1, x_2, \ldots, x_n$ and access to oracles $o_1, o_2, \ldots, o_n$. When X is a finite set, we use $x \in_R X$ to denote that x is chosen from X uniformly at random, and use $|X|$ to denote the size of X. We say that a function f is negligible in a parameter k, if for every polynomial $p(k)$, there exists an integer K such that for all $k > K, f(k) < \frac{1}{p(k)}$. For simplicity, we write $f(k) = negl(k)$. If $f(k)$ is negligible, then we say $1 - f(k)$ is overwhelming.

2.2. Assumptions

We use asymmetric bilinear map groups of Type-3 [18] for our construction. $Setup(k)$ is a bilinear group generator that takes a security parameter k as input, and outputs curve parameters $params = (p, G_1, G_2, G_T, e, g_1, g_2, g_T)$ where:

- G_1, G_2 and G_T are three disjoint cyclic subgroups on an elliptic curve of Type-3.
- g_1, g_2 and g_T are generators of G_1, G_2 and G_T.
- e is an efficiently-computable bilinear pairing map $e : G_1 \times G_2 \to G_T$ that satisfies two properties:
  (1) Bilinearity: for all $a, b \in Z_p$, $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$.
  (2) No-degeneracy: $e(g_1, g_2) \neq 1$.

Definition 1 (*Bilinear Diffie–Hellman Variant (BDHV) Assumption [16]*)**.** Given $T = (p, G_1, G_2, G_T, e, g_1, g_2)$, for all p.p.t. adversary A, for every sufficiently large security parameter k and $a, b, c \in_R Z_p$ and $R \in_R G_T$, A's advantage $\varepsilon_{BDHV} = |Pr[A(T, g_1^a, g_2^{\frac{1}{a}}, g_2^{\frac{b}{a}}, g_1^c, e(g_1, g_2)^{bc}) = 1] - Pr[A(T, g_1^a, g_2^{\frac{1}{a}}, g_2^{\frac{b}{a}}, g_1^c, R)]| = negl(k)$.

Definition 2 (*Symmetric EXternal Diffie–Hellman (SXDH) Assumption [19]*)**.** Given $T = (p, G_1, G_2, G_T, e, g_1, g_2)$, for all p.p.t. adversary A, for every sufficiently large security parameter k and $a, b, c, d, r_1, r_2 \in_R Z_p$, A's advantage $\varepsilon_{SXDH} = max(\varepsilon_1, \varepsilon_2) = negl(k)$, where $\varepsilon_1 = |Pr[A(T, g_1^a, g_1^b, g_1^{ab}) = 1] - Pr[A(T, g_1^a, g_1^b, g_1^{r_1}) = 1]|$, and $\varepsilon_2 = |Pr[A(T, g_2^c, g_2^d, g_2^{cd}) = 1] - Pr[A(T, g_2^c, g_2^d, g_2^{r_2}) = 1]|$.

Definition 3 (*n-Parallel Decisional Diffie–Hellman ($PDDH_n$) Assumption [20]*)**.** Given $T = (p, G_1, G_2, G_T, e, g_1, g_2)$, for all p.p.t.