A Fast and Secure Scheme for Data Outsourcing in the Cloud

Yanjun Liu[1, 2], Hsiao-Ling Wu[3] and Chin-Chen Chang[2, 3]
[1]Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education,
School of Computer Science and Technology, Anhui University,
Hefei, 230039 – China
[e-mail: yjliu104@gmail.com]
[2] Department of Computer Science and Information Engineering, Asia University,
Taichung, 41354 – Taiwan
[e-mail: yjliu104@gmail.com, alan3c@gmail.com]
[3] Department of Information Engineering and Computer Science, Feng Chia University,
Taichung, 40724 – Taiwan
[e-mail: wuhsiaoling590@gmail.com, alan3c@gmail.com]
*Corresponding author: Chin-Chen Chang

Received March 14, 2014; revised May 19, 2014; accepted June 18, 2014; published August 29, 2014

Abstract

Data outsourcing in the cloud (DOC) is a promising solution for data management at the present time, but it could result in the disclosure of outsourced data to unauthorized users. Therefore, protecting the confidentiality of such data has become a very challenging issue. The conventional way to achieve data confidentiality is to encrypt the data via asymmetric or symmetric encryptions before outsourcing. However, this is computationally inefficient because encryption/decryption operations are time-consuming. In recent years, a few DOC schemes based on secret sharing have emerged due to their low computational complexity. However, Dautrich and Ravishankar pointed out that most of them are insecure against certain kinds of collusion attacks. In this paper, we proposed a novel DOC scheme based on Shamir's secret sharing to overcome the security issues of these schemes. Our scheme can allow an authorized data user to recover all data files in a specified subset at once rather than one file at a time as required by other schemes that are based on secret sharing. Our thorough analyses showed that our proposed scheme is secure and that its performance is satisfactory.

Keywords: Data outsourcing, cloud, secret sharing, security, computational complexity

This research was supported in part by the National Nature Science Foundation of China (grant number: 61202228) and the College Natural Science Key Project of Anhui Province of China (grant number: KJ2012A008).

http://dx.doi.org/10.3837/tiis.2014.08.008

1. Introduction

At the present time, managing data securely and effectively poses monumental challenges in the area of cryptography. Data outsourcing is a new paradigm in which data are stored onto a trusted, external service provider, such as a cloud storage server. Data outsourcing in the cloud (DOC) is a promising solution for data management because it offers three beneficial features, i.e., 1) it provides on-demand, high-quality service from shared resources; 2) it provides universal data access by data users regardless of their locations; and 3) it reduces the costs of hardware and software [1, 2].

As DOC becomes more and more popular, data owners are storing huge quantities of data in the cloud. However, personal information, transmitted emails, financial data, and other sensitive or confidential data may be disclosed to unauthorized users. Thus, protecting the confidentiality of such data has become a main security issue. One of the most extensively used methods to fulfill this security requirement is to encrypt sensitive data before outsourcing them to prevent unauthorized access. Thus, only authorized users can decrypt the encrypted, outsourced data and obtain the plaintext, and unauthorized users are incapable of acquiring any of the original data.

To date, two approaches for the encryption of outsourced data have been proposed in the literature, and the traditional approach is to use an asymmetric or symmetric cryptosystem to conceal the content of the original data [3-8]. In 2011, Lu and Tsudik [4] proposed a DOC scheme based on an asymmetric cryptosystem to enhance the privacy of data. The scheme prevents the cloud server from knowing any plaintext of the outsourced data. In addition, it provides the data owner with content-level access control. Later, Raykova et al. [5] proposed a two-level, access-control scheme for DOC using a combination of asymmetric and symmetric encryption according to different access policies. Their scheme provides security guarantees for both data owners and data users. Zhou et al. [6] introduced a tree-based, key-management scheme in the cloud outsourcing environment that allows data users to access outsourced data with different levels of access rights. In their scheme, an asymmetric cryptosystem is used to encrypt data in each node with one key and decrypt it with two other keys. To achieve effective utilization of encrypted, outsourced data in the cloud, Wang et al. [7] proposed a ranked searchable symmetric encryption scheme. They showed that their method is secure and preserves the privacy of the data. Recently, Giweli et al. [8] proposed a robust DOC mechanism that integrated asymmetric encryption, symmetric encryption, and the Chinese remainder theorem.

However, along with the explosive growth in the amount of outsourced data and in the number of data users, each user may be authorized to access only a particular subset of data during a certain period of time. This makes the efficiency of data retrieval a very urgent challenge. The aforementioned solutions are not efficient since the computational cost of asymmetric or symmetric encryption/decryption operations is extremely high. Thus, secret sharing is emerging as an approach for the encryption of outsourced data due to its low computational complexity. Unlike the aforementioned encryption/decryption operations, secret sharing does not depend on any encryption/decryption keys. In recent years, only a few researchers [9-11] have focused on the development of DOC schemes based on Shamir's secret sharing. In the schemes proposed in [9-11], a data file is divided into n pieces shared among n cloud storage servers; knowledge of any t or more pieces can be used to recover the file. The developers of these schemes have claimed that they are secure, but Dautrich and

Ravishankar [12] pointed out that all of the three schemes are vulnerable to the collusion attack in which any t colluding servers can recover all files outsourced in the cloud.

Therefore, the objective of the scheme described in this paper was to achieve security and high efficiency at the same time in the context of DOC. We propose a fast and secure DOC scheme based on the concept of Shamir's secret sharing. The contributions of our proposed protocol are listed below:

(1) A specified subset of files is split into two types of shares, i.e., 1) public shares outsourced to the cloud storage servers and 2) private shares shared with an authorized data user. The authorized data user can combine the private shares with the public shares sent by the cloud storage servers to reconstruct the original files based on Shamir's secret sharing.

(2) In our proposed scheme, an authorized data user can recover all files in a specified subset at once. However, in other schemes based on secret sharing, only one file can be obtained at one time.

(3) Our proposed scheme satisfies fundamental security requirements.

(4) Our proposed scheme is more efficient than other related schemes.

The rest of the paper is organized as follows. Section 2 addresses some background information. Section 3 describes the details of our proposed scheme. Security and performance analyses of our proposed protocol are given in Section 4, and our conclusions are presented in Section 5.

2. Preliminary Information

In this section, we briefly introduce some essential background information regarding DOC. First, we describe the entities that participate in a typical DOC scheme. Second, we specify the basic security requirements that a DOC scheme should satisfy. Third, we introduce the concept of Shamir's secret sharing, which is used as the main building block in the design of our proposed DOC scheme.

2.1 Definitions and Entities

Three different entities, i.e., the data owner, the data user, and the cloud storage server, are involved in a classic DOC scheme that can preserve the privacy of the data by managing access to confidential files. The definition and responsibility of each entity are demonstrated as follows:

(1) Data owner. The data owner possesses a collection of confidential and sensitive data files that he/she will outsource to the cloud storage servers. To preserve the privacy of the data, the data owner does not store these files directly on the cloud but usually encrypts them into ciphertext before outsourcing.

(2) Data user. Authorized data users can access the plaintext of the outsourced data. In particular, if a data user has the privilege of accessing a specified subset of data files maintained by the data owner, he/she can decrypt the outsourced ciphertext and obtain the original data.

(3) Cloud storage server. Cloud storage servers store the data that are outsourced by the data owner. When a data user submits a request for data retrieval to the cloud storage servers, the corresponding set of encrypted files is returned immediately to the data user. The most important properties of cloud storage servers are that they are honest but curious. That is, they deal with requests from authorized data users honestly and return the correct outsourced information. However, they are curious in that they try to extract

as much information as possible about the outsourced data, some of which they should not know.

2.2 Security Requirements

A DOC scheme must satisfy four fundamental security requirements, i.e., data confidentiality, data correctness, query privacy, and collusion-resistance.

(1) **Data confidentiality**. This security requirement contains two basic aspects, i.e., 1) the plaintext of the outsourced data must not be revealed to outside attackers and 2) since the cloud storage servers are honest but curious, they should be prevented from accessing the original data files from the outsourced data they have stored.

(2) **Data correctness**. Authorized data users can decrypt the outsourced ciphertext and derive the plaintext according to their access rights to confidential files. On the contrary, unauthorized users who do not have the right to access specified files cannot obtain any useful information about the plaintext.

(3) **Query privacy**. Neither the honest-but-curious cloud storage servers nor a malicious, outside attacker can confirm which files the data user wants to access through the data user's data query to the cloud storage servers.

(4) **Collusion-resistance**. The collusion attack means that, if multiple cloud storage servers collude, they can share the information they hold to obtain a confidential file that neither of them alone can access. A DOC scheme should resist this type of attack.

2.3 Shamir's Secret Sharing Mechanism

Since we used the secret sharing mechanism proposed by Shamir in 1979 [13] as a main block in the construction of our proposed scheme, this subsection will thoroughly introduce the concept and principles of Shamir's secret sharing.

Shamir's secret sharing is a practical tool for safeguarding keys in the field of cryptography in which a dealer partitions a secret into *n* shares distributed among *n* shareholders. Based on the Lagrange interpolating polynomial, *t* or more shareholders can contribute their shares and cooperatively recover the secret; however, if fewer than *t* shares are available, the shareholders are unable to reconstruct the secret. Therefore, such a scheme is also called (*t*, *n*) Shamir's secret sharing, denoted as (*t*, *n*)-SSS.

Assume that a dealer *D* wants to share a secret *s* among *n* shareholders, $\{u_1, u_2, \ldots, u_n\}$, in a (*t*, *n*)-SSS. As a result, the share generation procedure and the secret reconstruction procedure must be conducted, and they are described as follows.

Share generation procedure

Dealer *D* constructs the following polynomial:

$$f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{t-1} x^{t-1} \bmod p, \tag{1}$$

where p is a prime, t coefficients, $a_0, a_1, a_2, \ldots, a_{t-1}$, are in the finite field $GF(p)$, and the secret $s = a_0 = f(0)$. By choosing n random numbers x_i for $i = 1, 2, \ldots, n$, dealer D generates n shares as $S_i = f(x_i)$ for $i = 1, 2, \ldots, n$. Then, D issues share S_i to shareholder u_i.

Secret reconstruction procedure

In this procedure, t shareholders can release their shares, $\{S_1, S_2, \ldots, S_t\}$, to recover the polynomial $f(x)$ generated by the dealer *D* based on the Lagrange interpolating theorem [13] as follows:

$$f(x)=\sum_{j=1}^{t} S_j \prod_{m=1,m\neq j}^{t} \frac{x-x_m}{x_j-x_m} \bmod p. \quad (2)$$

Obviously, the secret *s* can be reconstructed immediately by $s=f(0)$.

From the above procedures, we can infer that at least *t* distinct points, i.e., (x_1,S_1), (x_2,S_2), …, (x_t,S_t), are needed to recover a polynomial of degree *t*-1 in (*t*, *n*)-SSS. This scheme has been proven to be unconditionally secure [13-15], and it is used extensively in many applications of information security, such as group key distribution protocols [16-18], group authentication [19], and data outsourcing systems [9-11]. The following example illustrates the execution of a (3, 3)-SSS.

Example 2.1 Given $\{a_0,a_1,a_2\}=\{3,2,1\}$, $p=53$, and $\{x_1,x_2,x_3\}=\{4,5,6\}$, recover the secret *s* using a (3, 3)-SSS.

In the share generation procedure, dealer *D* uses three coefficients, a_0,a_1, and a_2, to construct a second-degree polynomial $f(x)$ as $f(x)=3+2x+x^2 \bmod 53$, where the secret $s=a_0=3$. Then, *D* generates three shares, i.e., $S_1=f(x_1)=27$, $S_2=f(x_2)=38$, and $S_3=f(x_3)=51$, and sends S_i secretly to shareholder u_i. In the secret reconstruction procedure, shareholders u_1, u_2, and u_3 work together to recover the original polynomial $f(x)$ based on the Lagrange interpolation:

$$\begin{aligned} f(x) &= (27\cdot\frac{x-5}{4-5}\cdot\frac{x-6}{4-6}+38\cdot\frac{x-4}{5-4}\cdot\frac{x-6}{5-6}+51\cdot\frac{x-4}{6-4}\cdot\frac{x-5}{6-5}) \bmod 53 \\ &= [40\cdot(x^2-11x+30)+15\cdot(x^2-10x+24)+52\cdot(x^2-9x+20)] \bmod 53 \\ &= x^2+2x+3 \bmod 53. \end{aligned}$$

Therefore, the secret *s* can be reconstructed as $s=f(0)=3$.

3. Our Proposed Scheme

In this section, we propose a fast and secure DOC scheme based on the (*n*, *n*)-SSS mechanism. First, we outline the architecture of our proposed scheme, and, then, a detailed description is given.

3.1 Architecture of Our Proposed Scheme

Fig. 1 shows the architecture of our proposed scheme using SSS. Assume that the data owner has a collection of data files to protect. Authorized data users can access different subsets of files depending on their positions and responsibilities in an organization. However, confidential files cannot be disclosed to unauthorized users. In our proposed scheme, the data owner splits a specified subset of files into two types of shares, i.e., public shares and private shares. Public shares are outsourced to the cloud storage servers, while private shares are transmitted to an authorized data user. When an authorized data user wants to access a particular subset of files, he/she submits a request to the cloud storage servers for the outsourced public shares of the files. Upon retrieving the public shares from the cloud storage servers, the data user can combine her/his private shares with the public shares to reconstruct the original files based on the concept of SSS.

Our proposed scheme consists of two phases, i.e., 1) the construction phase and 2) the recovery phase. The phases are discussed in detail in Subsections 3.2 and 3.3, respectively.

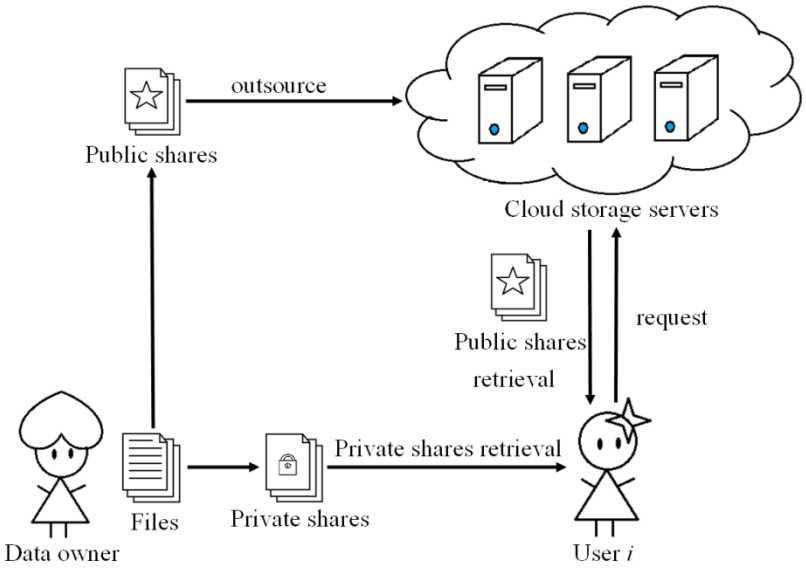


Fig. 1. Architecture of our proposed DOC scheme

3.2 Construction Phase

Assume that the data owner maintains a collection of *n* data files, $S=\{F_1, F_2, \cdots, F_n\}$. Let A_j represent a set of files that that an authorized data user can access. Thus, A_j is a subset of S. Assuming that the size of A_j is confined to $1\le|A_j|\le\left\lceil\frac{n}{2}-1\right\rceil$, there totally exists $(C_1^n+C_2^n+\cdots+C_{\left\lceil\frac{n}{2}-1\right\rceil}^n)$ A_j by different combinations of the files. Let $C=C_1^n+C_2^n+\cdots+C_{\left\lceil\frac{n}{2}-1\right\rceil}^n$, $U=\{A_j\}_{1\le j\le C}$, and each A_j is associated with a unique access number, $id(A_j)$. In addition, there are C cloud storage servers, $\{H_1, H_2, \cdots, H_C\}$.

In the construction phase, the data owner takes charge of generating *n* shares of each specified file subset A_j, including $|A_j|$ private shares and $(n-|A_j|)$ public shares. Then, the data owner outsources public shares on the corresponding cloud storage server H_j and shares private shares with the authorized data user. The construction phase is executed by the following steps:

Step 1. The data owner constructs a polynomial of degree *n*-1 as $f(x)=F_1+F_2x+\cdots+F_nx^{n-1} \bmod p$. Then, the data owner selects *n* random numbers c_i (also called the ***index*** of file F_i) and computes $S_i=f(c_i)$ for $i=1,2,...,n$. Here, (c_i, S_i) is considered as a ***private share*** of S.

Step 2. The data owner generates an additional polynomial $f_j(x)=\alpha_{j,1}+\alpha_{j,2}x+\cdots+\alpha_{j,n}x^{n-1} \bmod p$ of degree *n*-1 for each A_j, in which

$\alpha_{j,i} = F_i$ if $F_i \in A_j$ for $1 \le i \le n$.

Step 3. The data owner computes the remaining, unknown coefficients of $f_j(x)$ by the rules below. Let $f_j(x)$ pass through point (c_i, S_i) for all $F_i \in A_j$. Therefore, $|A_j|$ simultaneous equations with $(n-|A_j|)$ unknown coefficients of degree one are constructed. According to the simultaneous equations, a set of solutions of the unknown coefficients can be obtained in such a way that $\alpha_{j,k} \neq F_k$ if $F_k \notin A_j$ for $1 \le k \le n$.

Step 4. For all $F_k \notin A_j (1 \le k \le n)$, the data owner calculates $S_k = f_j(c_k)$ and then outsources $id(A_j)$ and the corresponding $(n-|A_j|)$ ***public shares*** (c_k, S_k) of A_j to the cloud storage server H_j.

3.3 Recovery Phase

Assume that the authorized data user have obtained $id(A_j)$ and $|A_j|$ ***private shares*** (c_i, S_i) of A_j satisfying $F_i \in A_j$ from the data owner. Thus, the authorized data user can combine private shares and the corresponding public shares to recover the original files in A_j that he/she wants to access.

Step 1. The authorized data user submits $id(A_j)$ to the cloud storage server H_j to request the outsourced public shares of A_j.

Step 2. H_j returns $(n-|A_j|)$ public shares of A_j to the data user.

Step 3. The data user uses $|A_j|$ private shares that he/she holds and the $(n-|A_j|)$ public shares received to recover $f_j(x)$ using the (n, n)-SSS mechanism. Then, the data user can easily retrieve A_j from $f_j(x)$.

Remark 1: In the proposed scheme, the size of Set A_j must be subject to the condition that $1 \le |A_j| \le \left\lceil \frac{n}{2} - 1 \right\rceil$. This is because there are $(n-|A_j|)$ unknown coefficients to be determined with $|A_j|$ simultaneous equations. If the size of Set A_j exceeds $\left\lceil \frac{n}{2} - 1 \right\rceil$, $|A_j| \ge (n-|A_j|)$ will hold, which indicates that the number of simultaneous equations is equal to or greater than that of unknown coefficients and the unique solution for these coefficients must be determined by setting $\alpha_{j,k} = F_k$ if $F_k \notin A_j$. Therefore, it will lead to the consequence that unauthorized users can retrieve the file F_k that they do not have the right to access from the relationship of $\alpha_{j,k} = F_k$ if $F_k \notin A_j$. This violates the concept of our proposed scheme. Otherwise, if $1 \le |A_j| \le \left\lceil \frac{n}{2} - 1 \right\rceil$, there are infinite possible solutions for unknown coefficients and an appropriate solution that satisfies $\alpha_{j,k} \neq F_k$ if $F_k \notin A_j$ for $1 \le k \le n$ can be selected.

Remark 2: In the construction phase, the data owner generates a polynomial $f_j(x)$ of degree n-1 that conceals all of the confidential files in A_j that the authorized data user can access. Then, the authorized data user can recover $f_j(x)$ successfully through the n shares he/she holds based on the (n, n)-SSS mechanism. Therefore, all files in A_j are obtained at once in our proposed scheme. However, in other related schemes, only one file at a time can be derived from A_j since a single polynomial contains only one file. This is the major advantage of our scheme over other schemes based on secret sharing, and it can lead to a higher efficiency.

3.4 Example

Assume that the data owner maintains five (n = 5) files, i.e., $F_1 = 10$, $F_2 = 9$, $F_3 = 8$, $F_4 = 7$, and $F_5 = 6$. Since there is a total of five files, an authorized user can access, at most, $\left\lceil \frac{5}{2} - 1 \right\rceil = 2$ files, which indicates that $|A_j| = 1$ or 2 and that $U = \{A_j\}_{1 \le j \le 15}$. More specifically, $U = \{A_1 = \{F_1\}, A_2 = \{F_2\}, A_3 = \{F_3\}, A_4 = \{F_4\}, A_5 = \{F_5\}, A_6 = \{F_1, F_2\}, A_7 = \{F_1, F_3\}, A_8 = \{F_1, F_4\}, A_9 = \{F_1, F_5\}, A_{10} = \{F_2, F_3\}, A_{11} = \{F_2, F_4\}, A_{12} = \{F_2, F_5\}, A_{13} = \{F_3, F_4\}, A_{14} = \{F_3, F_5\}, A_{15} = \{F_4, F_5\}\}$. We just choose how to access A_5 and A_6 as two scenarios to show the process of our proposed scheme.

Example 3.1

Construction phase

The data owner constructs the polynomial $f(x) = 10 + 9x + 8x^2 + 7x^3 + 6x^4 \bmod 31$. Then, he/she generates five private shares, $(c_1, S_1) = (1,9)$, $(c_2, S_2) = (2,26)$, $(c_3, S_3) = (3,29)$, $(c_4, S_4) = (4,19)$, and $(c_5, S_5) = (5,13)$.

The data owner also generates 15 additional polynomials. Among them, $f_5(x)$ and $f_6(x)$ are shown as follows:

$$\begin{aligned} f_5(x) &= \alpha_{5,1} + \alpha_{5,2}x + \alpha_{5,3}x^2 + \alpha_{5,4}x^3 + 6x^4 \bmod 31, \\ \text{and } f_6(x) &= 10 + 9x + \alpha_{6,3}x^2 + \alpha_{6,4}x^3 + \alpha_{6,5}x^4 \bmod 31, \end{aligned} \tag{3}$$

in which $f_j(x)$ contains the information of A_j. To determine the coefficients in (3), the data owner lets $f_5(x)$ pass through the point (c_5, S_5), and $f_6(x)$ pass through two points (c_1, S_1) and (c_2, S_2). Consequently, (3) becomes (4).

$$\begin{aligned} f_5 &: 14 = \alpha_{5,1} + 5\alpha_{5,2} + 25\alpha_{5,3} + \alpha_{5,4} \bmod 31, \\ \text{and } f_6 &: \begin{cases} 21 = \alpha_{6,3} + \alpha_{6,4} + \alpha_{6,5} \bmod 31, \\ 29 = 4\alpha_{6,3} + 8\alpha_{6,4} + 16\alpha_{6,5} \bmod 31. \end{cases} \end{aligned} \tag{4}$$

Because $\alpha_{jk} \neq F_k$ for $1 \le j \le 15$ and $1 \le k \le 5$, an appropriate solution of the unknown coefficients is listed as follows: $\alpha_{5,1} = 7$, $\alpha_{5,2} = 11$, $\alpha_{5,3} = 14$, $\alpha_{5,4} = 5$, $\alpha_{6,3} = 12$, $\alpha_{6,4} = 1$, and $\alpha_{6,5} = 8$. After that, the data owner outsources $id(A_j)$ and public shares of A_j to the

cloud storage server H_j, where $1 \le j \le 15$. The data owner also sends $id(A_j)$ and private shares of A_j to the authorized data user. **Table 1** lists the crucial information associated with A_5 and A_6 in the construction phase.

Recovery phase

Assume that an authorized data user wants to access $A_5 = \{F_5 = 6\}$ and that he/she submits $id(A_5)$ to the cloud storage server H_5. H_5 returns four public shares of A_5 (as shown in **Table 1**) to the data user. Finally, the data user uses private share (5, 13) and public shares to recover polynomial $f_5(x)$ as follows:

$$\begin{aligned}
f_5(x) &= \sum_{j=1}^{5} S_j \prod_{m=1, m\neq j}^{5} \frac{x - c_m}{c_j - c_m} \bmod p \\
&= (12 \cdot \frac{x-2}{1-2} \cdot \frac{x-3}{1-3} \cdot \frac{x-4}{1-4} \cdot \frac{x-5}{1-5} + 4 \cdot \frac{x-1}{2-1} \cdot \frac{x-3}{2-3} \cdot \frac{x-4}{2-4} \cdot \frac{x-5}{2-5} \\
&\quad + 12 \cdot \frac{x-1}{3-1} \cdot \frac{x-2}{3-2} \cdot \frac{x-4}{3-4} \cdot \frac{x-5}{3-5} + 23 \cdot \frac{x-1}{4-1} \cdot \frac{x-2}{4-2} \cdot \frac{x-3}{4-3} \cdot \frac{x-5}{4-5} \\
&\quad + 13 \cdot \frac{x-1}{5-1} \cdot \frac{x-2}{5-2} \cdot \frac{x-3}{5-3} \cdot \frac{x-4}{5-4}) \bmod 31 \\
&= (16 \cdot (27 + x + 9x^2 + 17x^3 + x^4) + 20 \cdot (29 + 17x + 28x^2 + 18x^3 + x^4) \\
&\quad + 3 \cdot (9 + 15x + 18x^2 + 19x^3 + x^4) + 22 \cdot (30 + x + 10x^2 + 20x^3 + x^4) \\
&\quad + 7 \cdot (24 + 12x + 4x^2 + 21x^3 + x^4)) \bmod 31 \\
&= 7 + 11x + 14x^2 + 5x^3 + 6x^4 \bmod 31.
\end{aligned}$$

Therefore, the data user is able to obtain $F_5 = 6$. Similarly, if an authorized data user wants to access $A_6 = \{F_1 = 10, F_2 = 9\}$, he/she can reconstruct $f_6(x)$ by using private and public shares (as shown in **Table 1**) as follows:

$$\begin{aligned}
f_6(x) &= \sum_{j=1}^{5} S_j \prod_{m=1, m\neq j}^{5} \frac{x - c_m}{c_j - c_m} \bmod p \\
&= (9 \cdot \frac{x-2}{1-2} \cdot \frac{x-3}{1-3} \cdot \frac{x-4}{1-4} \cdot \frac{x-5}{1-5} + 26 \cdot \frac{x-1}{2-1} \cdot \frac{x-3}{2-3} \cdot \frac{x-4}{2-4} \cdot \frac{x-5}{2-5} \\
&\quad + 14 \cdot \frac{x-1}{3-1} \cdot \frac{x-2}{3-2} \cdot \frac{x-4}{3-4} \cdot \frac{x-5}{3-5} + 25 \cdot \frac{x-1}{4-1} \cdot \frac{x-2}{4-2} \cdot \frac{x-3}{4-3} \cdot \frac{x-5}{4-5} \\
&\quad + 24 \cdot \frac{x-1}{5-1} \cdot \frac{x-2}{5-2} \cdot \frac{x-3}{5-3} \cdot \frac{x-4}{5-4}) \bmod 31 \\
&= (12 \cdot (27 + x + 9x^2 + 17x^3 + x^4) + 6 \cdot (29 + 17x + 28x^2 + 18x^3 + x^4) \\
&\quad + 19 \cdot (9 + 15x + 18x^2 + 19x^3 + x^4) + 1 \cdot (30 + x + 10x^2 + 20x^3 + x^4) \\
&\quad + 1 \cdot (24 + 12x + 4x^2 + 21x^3 + x^4)) \bmod 31 \\
&= 10 + 9x + 12x^2 + x^3 + 8x^4 \bmod 31.
\end{aligned}$$

Therefore, the data user is able to obtain $F_1 = 10$ and $F_2 = 9$.

Table 1. Information associated with A_5 and A_6 in Example 3.1

Access number	$id(A_5)$	$id(A_6)$
File subset A_j	$A_5 = \{F_5 = 6\}$	$A_6 = \{F_1 = 10, F_2 = 9\}$
Polynomial $f_j(x)$	$f_5(x) = 7 + 11x + 14x^2 + 5x^3 + 6x^4 \bmod 31$	$f_6(x) = 10 + 9x + 12x^2 + x^3 + 8x^4 \bmod 31$
Public shares of A_j **on** H_j	$(1,12)$, $(2,4)$, $(3,12)$, $(4,23)$	$(3,14)$, $(4,25)$, $(5,24)$
Private shares of A_j	$(5,13)$	$(1,9)$, $(2,26)$

4. Analyses

In this section, we analyze the security and performance of our proposed scheme. First, we show that our proposed scheme can achieve fundamental security requirements. Then, the performances of our proposed scheme and other related schemes are compared.

4.1 Security Analysis

Here, we show that our proposed scheme can satisfy four fundamental security requirements, i.e., data confidentiality, data correctness, query privacy, and collusion-resistance.

(1) Data confidentiality

Our proposed scheme can ensure the confidentiality of the data based on the fact that *n* shares must be collected to recover the secret in the (*n*, *n*)-SSS. Assume that *l* represents the size of a particular subset of files, A_j, that an authorized data user wants to access. In our proposed scheme, the data owner outsources only $(n-l)$ public shares of this subset of files on the cloud storage server H_j. Therefore, the curious H_j cannot recover the original subset of files through the stored public shares since it does not have the other *l* private shares shared by the data owner and the data user. Furthermore, even if an outside attacker intrudes on the cloud storage servers and observes public shares, he/she cannot obtain the correct files for the same reason.

(2) Data correctness

If a data user is authorized to access a certain subset of files, he/she submits the access number of this subset to the corresponding cloud storage server to request $(n-l)$ outsourced public shares of this subset. Upon receiving the public shares, the user can recover the correct subset of files by combining the *l* private shares that were obtained from the data owner secretly in the construction phase with $(n-l)$ public shares based on (*n*, *n*)-SSS. Therefore, in our proposed scheme, the correctness of (*n*, *n*)-SSS determines the correctness of the data. However, if a data user attempts to retrieve the files that he/she does not have the right to access, he/she will learn nothing about the files because he/she does not know their private shares.

(3) Query privacy

In our proposed scheme, each A_j that contains file information is assigned an access

number, i.e., $id(A_j)$, and the data user submits $id(A_j)$ to request public shares of A_j. Since the public shares of A_j have no direct relationship with A_j and do have such a relationship with $id(A_j)$, neither a cloud storage server nor an outside attacker can determine the subset of files to which these public shares belong. Thus, query privacy is achieved in our proposed scheme.

(4) Collusion-resistance

In our method, only $(n-l)$ public shares of a specified subset of files are outsourced to a cloud storage server. Thus, each cloud storage server cannot recover a polynomial of degree n-1 to obtain all files in the subset at once, since these shares are not sufficient. If multiple cloud storage servers collude, they still cannot obtain any file. This is because different cloud storage servers store public shares of different subsets of files; even if they exchange their shares, they cannot collect enough shares of a single subset. Therefore, a collusion attack cannot be launched successfully.

Comparisons of the security provided by our proposed scheme and other SSS-based schemes [9-11] are provided in **Table 2**. From the results in **Table 2**, we can infer that our proposed scheme can satisfy all security requirements mentioned in Subsection 2.2, while the other schemes are unable to resist collusion attacks [12].

Table 2. Comparisons among different schemes

	Security property				Communication cost		Type
	Data confidentiality	Data correctness	Query privacy	Collusion-resistance	Numbers of public shares	Rounds of transmission	
[9]	Yes	Yes	Yes	No	$l \times t$	l	(t,n) SSS-based
[10]	Yes	Yes	Yes	No	$l \times t$	l	(t,n) SSS-based
[11]	Yes	Yes	Yes	No	$l \times t$	l	(t,n) SSS-based
Ours	Yes	Yes	Yes	Yes	$n-l$	1	(n,n) SSS-based

4.2 Performance Analysis

In this section, we evaluate the performance of our proposed scheme. According to the introduction described in Section 1, so far, there are two types of DOC schemes in the literature, one based on asymmetric or symmetric cryptosystem and the other based on SSS. One symmetric cryptosystem (DES) was about 100 times faster than one asymmetric cryptosystem (RSA-1024), and one SSS is 26 times faster than one symmetric cryptosystem [20]. Therefore, the computational cost of the schemes based on SSS is much lower than that of schemes based on asymmetric or symmetric cryptosystems. **Table 2** compares the performances of our proposed scheme and other schemes [9-11], all of which are based on SSS. Since there is no significant difference in the computational costs of these schemes, we only compare the communication cost in **Table 2**. Let l denote the size of a particular subset of files that an authorized data user wants to access, where $l \le \left\lceil \frac{n}{2} - 1 \right\rceil$. **Table 2** shows that recovering all l files in a subset requires $l \times t$ public shares in l rounds of transmission in the schemes proposed in [9-11], while our proposed scheme requires only $n-l$ public shares at once to complete the same task. Thus, our proposed scheme can reduce the communication cost significantly compared to other schemes.

5. Conclusions

In this paper, we proposed a novel DOC scheme based on Shamir's secret sharing mechanism. Unlike other DOC schemes based on secret sharing, our scheme can allow an authorized data user to recover all data files in a specified subset at once rather than one file at a time. Our proposed scheme can achieve all basic security requirements, such as data confidentiality, data correctness, query privacy, and collusion-resistance. Performance analyses demonstrated that the performance of our proposed scheme exceeded the performances of other related schemes.

Acknowledgements

This research was supported in part by the National Nature Science Foundation of China (grant number: 61202228) and the College Natural Science Key Project of Anhui Province of China (grant number: KJ2012A008).

References

[1] M. N. O. Sadiku, S. M. Musa and O. D.Momoh, "Cloud computing: opportunities and challenges," *IEEE Potentials*, vol. 33, no. 1, pp. 34-36, 2014. Article(CrossRefLink)

[2] M. Armbrust, A.Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin and M. Zaharia, "Above the clouds: a berkeley view of cloud computing," *University of California, Berkeley, Technical Report No. UCB/EECS-2009-28*, Feb. 2009.

[3] J. Hur and D. K. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 7, pp. 1214-1221, 2011. Article(CrossRefLink)

[4] Y. Lu and G. Tsudik, "Enhancing data privacy in the cloud," *Proceedings of IFIP Advances in Information and Communication Technology*, Copenhagen, Denmark, pp. 117-132, 2011. PMCid:PMC3630519

[5] M. P. Raykova, S. M. Bellovin and H. Zhao, "Privacy enhanced access control for outsourced data sharing," *Proceedings of Financial Cryptography and Data Security*, Kralendijk, Bonaire, pp. 223-238, Mar. 2012. Article(CrossRefLink)

[6] M. Zhou, Y. Mu, W. Susilo, J. Yan and L. Dong, "Privacy enhanced data outsourcing in the cloud," *Journal of Network and Computer Applications*, vol. 35, no. 4, pp. 1367-1373, 2012. Article(CrossRefLink)

[7] C. Wang, N. Cao, K. Ren and W. Lou, "Enabling secure and efficient ranked keyword search over outsourced cloud data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 8, pp. 1467-1479, 2012. Article(CrossRefLink)

[8] N. Giweli, S. Shahrestani and H. Cheung, "Enhancing data privacy and access anonymity in cloud computing," *Communications of the IBIMA*, article in press, 2013, Article(CrossRefLink)

[9] M. A. Hadavi and R. Jalili, "Secure data outsourcing based on threshold secret sharing; towards a more practical solution," in *Proc. of Proceedings of the 36th International Conference on Very Large Data Bases*, Singapore, pp. 54-59, Sep. 2010.

[10] D. Agrawal, A. A. El, F. Emekci, A.Metwally and S. Wang, "Secure data management service on cloud computing infrastructures," *Proceedings of Service and Application Design Challenges in the Cloud*, pp. 57-80, 2011.

[11] X. Tian, C. Sha, X. Wang and A. Zhou, "Privacy preserving query processing on secret share based data storage," in *Proc. of Proceedings of the 16th International Conference on Database Systems for Advanced Applications*, Hong Kong, China, pp. 108-122, Apr. 2011. Article(CrossRefLink)

[12] J. L. Dautrich and C. V. Ravishankar, "Security limitations of using secret sharing for data outsourcing," in *Proc. of Proceedings of the 26th Annual IFIP WG 11.3 Conference on Data and Applications Security and Privacy*, Paris, France, pp. 145-160, Jul. 2012.

[13] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, 1979. Article(CrossRefLink)

[14] G. R. Blakley, "Safeguarding cryptographic keys," *Proceedings of American Federation of Information Processing Societies National Computer Conference*, New York, USA, vol. 48, pp. 313-317, Nov. 1979.

[15] L. Harn and C. Lin, "Strong (*n*, *t*, *n*) verifiable secret sharing scheme," *Information Sciences*, vol. 180, no. 16, pp. 3059-3064, 2010. Article(CrossRefLink)

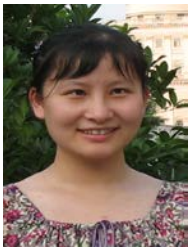
[16] C. Guo and C. C. Chang, "An authenticated group key distribution protocol based on the generalized Chinese remainder theorem," *International Journal of Communication Systems*, article in press, 2012. Article(CrossRefLink)

[17] L. Harn and C. Lin, "Authenticated group key transfer protocol based on secret sharing," *IEEE Transactions on Computers*, vol. 59, no. 6, pp. 842-846, 2010. Article(CrossRefLink)

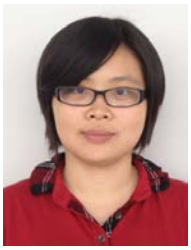
[18] Liu Y., L. Harn and C. C. Chang, "An authenticated group key distribution mechanism using theory of numbers," *International Journal of Communication Systems*, article in press, 2013, DOI: 10.1002/dac.2569. Article(CrossRefLink)

[19] L. Harn, "Group authentication," *IEEE Transactions on Computers*, vol. 62, no. 9, pp. 1893-1898, 2013. Article(CrossRefLink)

[20] Schneier B., *Applied cryptography, protocols, algorithms, and source code in C*, 2nd Edition, John Wiley and Sons Inc., New York, U.S.A., 1996.



Yanjun Liu received her B.S. degree in 2005, in School of Computer Science and T echnology from Anhui University, Hefei, China. She received her Ph.D. degree in 20 10, in School of Computer Science and Technology from University of Science and T echnology of China, Hefei, China. She is currently serving in Anhui University. Mea nwhile, she is a postdoctor at Asia University, Taichung, Taiwan. Her current research interests include information security and computer cryptography.



Hsiao-Ling Wu was born in Kaohsiung, Taiwan, in 1986. She received the BS degree in Applied Mathematics from Feng Chia University, Taichung, Taiwan in 2007. She is currently pursuing her Ph.D. degree in information engineering and computer science from Feng Chia University, Taichung, Taiwan. Her current research interests include electronic commerce, information security, cryptography, and mobile communications.

Chin-Chen Chang received his Ph.D. degree in computer engineering from National Chiao Tung University. His first degree is Bachelor of Science in Applied Mathematics and master degree is Master of Science in computer and decision sciences. Both were awarded in National Tsing Hua University. Dr. Chang served in National Chung Cheng University from 1989 to 2005. His current title is Chair Professor in Department of Information Engineering and Computer Science, Feng Chia University, from Feb. 2005. Prior to joining Feng Chia University, Professor Chang was an associate professor in Chiao Tung University, professor in National Chung Hsing University, chair professor in National Chung Cheng University. He had also been Visiting Researcher and Visiting Scientist to Tokyo University and Kyoto University, Japan. During his service in Chung Cheng, Professor Chang served as Chairman of the Institute of Computer Science and Information Engineering, Dean of College of Engineering, Provost and then Acting President of Chung Cheng University and Director of Advisory Office in Ministry of Education, Taiwan. Professor Chang has won many research awards and honorary positions by and in prestigious organizations both nationally and internationally. He is currently a Fellow of IEEE and a Fellow of IEE, UK. And since his early years of career development, he consecutively won Outstanding Talent in Information Sciences of the R. O. C., AceR Dragon Award of the Ten Most Outstanding Talents, Outstanding Scholar Award of the R. O. C., Outstanding Engineering Professor Award of the R. O. C., Distinguished Research Awards of National Science Council of the R. O. C., Top Fifteen Scholars in Systems and Software Engineering of the Journal of Systems and Software, and so on. On numerous occasions, he was invited to serve as Visiting Professor, Chair Professor, Honorary Professor, Honorary Director, Honorary Chairman, Distinguished Alumnus, Distinguished Researcher, Research Fellow by universities and research institutes. His current research interests include database design, computer cryptography, image compression and data structures.