PPDS: Privacy Preserved Data Sharing Scheme for Cloud Storage

Shu Qin Ren, Khin Mi Mi Aung

Abstract

When enterprises outsource their data to third-party cloud storage providers, they can save storage infrastructure investments and furthermore reduce administration costs tremendously. However data privacy and integrity are hindering their data migration steps to flexible and cost effective cloud storage. And traditional security schemes can provide data protection for this complicated cloud environment by sacrificing convenient operations, such as searching and sharing, which conflicts the flexibility and availability of cloud environment with multi-services for multi-tenants. This raises the demand for innovative and secure searching schemes to protect data privacy, data storage and retrieval process, and to access data flexibly on cloud storage. In this paper, we proposed a privacy preserved data sharing scheme on cloud storage to protect data privacy without sacrificing the cloud flexibility and accessibility: 1) Privacy Preserving Data Searching scheme to provide secure and efficient searching and sharing on privacy hidden data on cloud storage; 2) Policy based access control provides flexible yet data access ability on the privacy protected data.

Keywords: Privacy Protection on Cloud Storage, Private Keywords Searching And Retrieval, Private Data Sharing

1. Introduction

Gartner ranked cloud computing as the top 1 CIO technology in 2011 [12] because of its obvious flexibility and availability at low cost, where cloud storage plays as a big role in this trend by providing storage as a service. But security and privacy issues are hindering the data migration to the cloud. Traditionally data storage servers must be fully trusted, data and the infrastructure are both owned and managed by organizations themselves, the access of data is not revealed without authorization. When data is migrated to the cloud storage, cloud infrastructure is mostly owned and managed by service provider, yet data is spreading all over the places and the access of data is not completely under the control of users, which is not trusted as the traditional storage environment. We need a flexible and fine-grained cryptographic access control of data for users' convenience without leaking their privacy.

Cryptography is commonly used to protect data at rest which also can be applied for data protection on cloud. However with this layer of protection, the conventional document searching by indexed keywords is not compatible anymore because the conventional index constructions such as those using hash tables are not suitable for indexing sensitive data which leak information about the contents. Public-key Encryption with Keyword Search (PEKS) introduced by Boneh [2] is the first practical asymmetric searchable encryption scheme. Researchers also identified a few other security and privacy issues relevant to PEKS [2, 3, 4, 5, 13]. One major concern is to limit the delegate's capability of keyword searching within certain time frame [2, 3]. Another important concern is that PEKS is subject to offline keyword guessing attacks firstly identified by Byun [4]. Some research is concerning on the real time security and privacy such as the real-time application in mobile devices [14]. However these schemes didn't concern the large shared storage situation, where data sharing and searching are among chunks of data.

In this paper, we demonstrate how to provide efficient private keyword searching and sharing atop of today's cloud storage without compromising data confidentiality. The scheme provides the following special attributes:

- The schemes decouple the encrypted data and encryption keys. The encryption keys are managed by organizer itself based on fine-grained policies. The encrypted data are managed by third cloud storage providers.
- The schemes ensure that storage provider can provide fast speed searching ability without compromising data privacy with binary index tree.
The schemes ensure that storage providers provide flexible data sharing ability across cloud for group, subgroup or individual members.

2. Private Keywords Search and Retrieval

The sensitive or private data is normally protected by encryption. However, it may hinder further data searching and data sharing process. Researchers have recently investigated Private Information Retrieval (PIR), Search on Encrypted Data and Secure Function Evaluation (SFE) [1, 6, 7, 9] to support privacy protection when data is outsourced to the untrusted storage. With these references, we abstract the private keywords search and retrieval processes into the following 3 steps and depict the processes with Figure 1. All the notations used in this paper are also listed in Table 1.

![Figure 1. Private Keywords Search and Retrieval](image)

### Table 1. Notations and Definitions Used in the System

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Wi)</td>
<td>the i-th keyword in cleartext.</td>
</tr>
<tr>
<td>(K_d)</td>
<td>the data encryption key.</td>
</tr>
<tr>
<td>(K_c)</td>
<td>the control key.</td>
</tr>
<tr>
<td>(K_a)</td>
<td>the authorization key.</td>
</tr>
<tr>
<td>(Xi)</td>
<td>the i-th keyword after pre-encryption, composing of two parts: Li and Ri.</td>
</tr>
<tr>
<td>(Li)</td>
<td>the left ((n- m))-bit of (Xi).</td>
</tr>
<tr>
<td>(Ri)</td>
<td>the right (m)-bit of (Xi).</td>
</tr>
<tr>
<td>(Si)</td>
<td>the random number attached to (Wi), which is also one of the encryption key elements.</td>
</tr>
<tr>
<td>(ki)</td>
<td>the Hash key used to hash the random number (Si), generated by one way function (f_{k0}(Li)).</td>
</tr>
<tr>
<td>(Ci)</td>
<td>ciphertext of the i-th keyword on cloud.</td>
</tr>
<tr>
<td>(CiR)</td>
<td>index tag for the i-th pre-encrypted keyword.</td>
</tr>
<tr>
<td>(KDi)</td>
<td>the data encryption key.</td>
</tr>
<tr>
<td>(KCi)</td>
<td>the control key which is used to encrypt the encryption key.</td>
</tr>
<tr>
<td>(Pi)</td>
<td>the policy related to key word (Wi).</td>
</tr>
</tbody>
</table>

### Privacy Protection by Data Encryption

Assume a secure random generator \(G\) generate random number \(Si\) of \((n-m)\) bit long and a secure hash function \(H_{Li}(Si)\) on hash key \(k_o\), the sequence of tuple \(T_i = (Si, H_{Li}(Si))\) is a secure pseudo
random generator. For a screened sensitive data $W_i$, it has to be protected yet still can be searchable. The privacy protection work flow is depicted in Fig.1(A). Given a private keyword $W_i$, it pre-encrypted using a deterministic encryption algorithm $E_k(.)$ with data encryption key $k$, as $X_i = E_k(W_i)$. This data is further split into $n$-bit $L_i$ and $m$-bit $R_i$ respectively. For randomness, $<L_i, R_i>$ are operated by xor to the above tuple $T_i = <S_i, H_{k_i}(S_i)>$ with ciphertext output of $C_i = X_i \oplus T_i$. Here, we let $k_i = f_{k_i}(L_i)$.

- **Private Keywords Search.**

Now data owner wants to search keywords without revealing the words he is searching, so the data owner hides the keyword he wanted to search by pre-encryption, as shown in Fig.1(B). Now the hidden keyword to search is $X_i$. Third-party storage provider has to scan and search the user specific data without knowing the exact data, as depicted in Fig.1(C). The storage provider split the received data $X_i$ into two segments with $n$-bit $L_i$ and $m$-bit $R_i$. From $L_i$, the hash key $k_i = f_{k_i}(L_i)$ can be calculated. Next, $<L_i, R_i>$ will go for xor operation with the stored ciphertext $C_i$ and generate a $n$-bit output with $(n-m)$-bit $L_i'$ and $m$-bit $R_i'$. If $R_i' = H_{k_i}(L_i)$, the data exists in the scanned data, otherwise it does not exist there.

- **Private Keywords Retrieval.**

As a legitimate user with data encryption key $(S_i, k)$, he can decrypt data back. The work ow of the decryption can be referred from Fig.1(D). For the existing hidden keywords $C_i$ in the context, it goes to xor with the previously generated random number $S_i$ and it can get first $(n-m)$ bit as $L_i$. With $L_i$, it also can generate the hash key $K_i$ by $K_i = f_{k_i}(L_i)$. With $S_i$ and $K_i$, the storage provider also can hash $S_i$ by $H_{k_i}(S_i)$. The generated hash value goes with the lowest $m$-bit, it can recover $R_i$, the contact of $L_i$ and $R_i$ is $X_i$. The plaintext of this keyword is recovered after the corresponding decryption as $W_i = D_k(X_i)$.

### 3. A Privacy Protected Data Searching and Sharing Scheme on Cloud Storage

While data owners are outsourcing data to the third-party cloud storage services, they are hindered by data security and privacy risks. They have to classify and protect data privacy against data leakage before outsourcing them to the cloud. However strict data protection may affect the flexible usage of cloud storage, such as data retrieval, data searching, and data sharing and so on. Especially data searching on the privacy assured data will be a challenge. How would the data owner preserve data privacy without sacrificing the flexibility of cloud storage service as if the data were plaintext on trusted storage? With these considerations, we proposed a privacy protected data searching and sharing scheme for cloud storage.

#### 3.1. Three Main Players in the System

For flexible but privacy assured operation on the rented cloud storage platform, there are three parties in the system:

1) **Data Owner.** The entity who originates data to be stored on the cloud can retrieve and query data. The owner can interact with Key Manager to negotiate the authentication and authorization by specifying policy. Each policy is corresponding a control key $K_c$. And the private data will be encrypted by data encryption key $k$ which will be protected by the control key as $E_{K_c}(k)$.

2) **Key Manager.** It manages data authentication and authorization based on policy bound by data owner. This policy-based authorization will further generate a key $K_s$ which is sent to Data Owner for wrapping the control key by $E_{K_s}(K_c)$.

3) **Cloud Storage Provider.** A third-party storage service provider (e.g., Amazon S3) is providing data hosting service. When data owner uploads his data to cloud storage, he will upload the privacy protected data, policy, wrapped control key, wrapped encryption key as well.
3.2. Privacy Preserving Data Searching and Sharing on Cloud Storage

Figure 2. The Privacy-preserving Architecture of Data Hosting on Cloud Storage

The data owner would protect data privacy from eavesdropping and cloud storage while he put data on untrusted storage cloud and wish to enjoy the flexibility of cloud. It requires that storage cloud provider can host data and further provide data searching and data sharing functions without violating data privacy. The properties of privacy preserving data searching and data sharing make this scheme suitable for data hosting service with privacy assurance. And the functions are implemented by two agents: data owner agent \(A_{\text{data.owner}}\) and cloud storage agent \(A_{\text{cloud:storage}}\) respectively. The architecture of this scheme is illustrated in Fig. 2. Based on the Private Keywords Searching literatures, we emphasize the searching speed and secure privacy sharing on large scale shared storage, such as cloud storage by adding index tag to support dynamic sorting tree and flexible policy-based access control. This scheme will fulfill the following three main functions, with the depiction of Fig 3.

- Privacy Preserving on the Cloud.

  Data owner would preserve privacy against overhearing and data hostage by its agent \(A_{\text{data.owner}}\) using the privacy protection mechanism such as pre-encryption and permutation as demonstrated in section 2 and Fig. 1(A). This agent is in charge of privacy filtering and privacy preserving: 1) A pre-regulated privacy criteria pool is defined by data owner; 2) Based on this rule set, the data owner agent \(A_{\text{data.owner}}\) screens sensitive data and protects these data by pre-encryption; 3) The data owner agent \(A_{\text{data.owner}}\) creates hidden keywords set as additional metadata; 4) \(A_{\text{data.owner}}\) sends the privacy preserved data with additional metadata to the cloud storage. Except the privacy protected key words, an index tag is attached to each key word as well, as \(C_i\) indicated in Fig 3(A).

- Secure Data Searching across the Cloud.

  Data owner can send data query to the storage cloud without revealing the data value. The cloud provider has the ability to scan and check the existence of data even the provider does not know the data value. As SFE enables distrustful parties compute a function without revealing its input to the other parties, it provides private information search ability on the cloud, as illustrated in Fig 1(C). Yet with big chunks of data on cloud, data searching efficiency have big impact on the data retrieval performance. An efficient private information search engine is demanded to load on the cloud storage agent to meet this demand.
To improve the search efficiency on the hidden data, we embed a private index tree to the cloud agent ($A_{cloud, storage}$) to enhance the private information retrieval ability on cloud. The existing private data retrieval schemes cannot provide data sorting or indexing by hashing. The private data is not order preserving at all. For this, we add index tag ($CiR$) which is the last $m$-bit $R_i$ of each private keyword as additional metadata before sending to the cloud storage. This tag is just partial of the ciphertext of the private keyword, it cannot disclose any content of the keyword but it can help build an index tree. With this private index tree built, the searching process can speed up tremendously. Furthermore, this index tree will not release any information except the order of those partial ciphertexts and it is dynamically maintained by storage cloud provider to keep the tree balance.

![Figure 3. The Privacy-preserving Key Words Search on Cloud Storage: (A) Data Privacy Protection by Data Owner; (B) Hidden Keywords Index Tree on Cloud provider; (C) Secure Existence Evaluation on Cloud Provider.](image)

- **Secure Data Sharing across the Cloud.**

  Secure data sharing is an extended feature from the above two; the private data can be shared among a group of legitimate users from cloud storage. How could the cloud storage provide a data sharing platform with flexible data retrieval and searching functions without sacrifice privacy assurance?

  We separate the access control and data encryption, as depicted in Fig.4. First, the access control is based on policy which will be charged by key manager to generate the authorization key $K_a$ to the data owner. Second, the data owner $U_i$ will be responsible for the encryption key $K_i$ and the control key $K_c$ to protect the private data and encryption key respectively. Third, the data owner sends cloud storage the following information: privacy protected data $F'$ including
public and private data, wrapped encryption key $E_{K_c}(K_d)$, wrapped control key $E_{K_d}(K_c)$. The similar mechanism was proposed in paper [8].

![Diagram of PPDS: Privacy Preserved Data Sharing Scheme for Cloud Storage](image)

**Figure 4.** Private Data Sharing among users on Cloud Storage

With the above information on cloud storage, when the other legitimate user $U_2$ accesses the cloud storage, he cannot get $U_1$'s private data without the authorization from the Key Manager. When he talks to Key Manager with his identification or attributes, Key Manager can validate his authentication and authorization, further light communication between them can let user $U_2$ securely get authorization key $K_a$ using blinded RSA [8]. Here $K_a$ ensures $U_2$ get $K_c = D_{K_d}(E_{K_d}(K_c))$ and further get $K_d = D_{K_c}(E_{K_c}(K_d))$. With the data key $K_d$, $U_2$ is able to retrieval private data back.

With this mechanism, when the policy is revoked or changed, it only affects the control key $K_c$ and the encryption of data key $E_{K_c}(K_d)$, but it does not affect the data key $K_d$ and the private keyword on cloud $C_i = E_{K_c}(W_i) \oplus <S_i, H_{K_d}(S_i)>$. This shows very flexible access control with policy bound, policy renewal and policy revocation. Besides, it also redirects the headache encryption key management to flexible control key management. With this scheme, we provide light weight communication by adding searching and sharing ability on cloud.

4. Conclusion: We Can Get Cloud with Privacy Preserved

The proposed scheme can mitigate the privacy leakage on cloud storage, which may motivate the cloud storage service providers to provide data hosting services for enterprises which consider data privacy as their assets. Secondly, the proposed scheme uses policy based access control to provide flexible data sharing across cloud at light communication on enterprise side. Thirdly, the proposed scheme separates the key management and data encryption to provide a secure yet lightweight re-keying process. In summary, the proposed scheme provides a secure data hosting service for cloud storage providers and flexible data sharing management for enterprises at light computation cost. It may potentially push the wider adoption of cloud storage usage for enterprises to reduce storage cost and administration fee tremendously, yet still with data privacy and management guaranteed.
5. References