Journal of Universal Computer Science, vol. 21, no. 3 (2015), 440-453
submitted: 30/6/14, accepted: 28/2/15, appeared: 1/3/15 © *J.UCS*

Searchable Public-Key Encryption with Data Sharing in Dynamic Groups for Mobile Cloud Storage

Qi Xia, Jianbing Ni
(Big Data Research Center, School of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu, 611731, China
xiaqi@uestc.edu.cn, nimengze@gmail.com)

Ansuura John Bosco Aristotle Kanpogninge
(University for Development Studies, Box TL1360, Tamale, Ghana
jansuura@gmail.com)

James C. Gee
(School of Engineering and Applied Science, University of Pennsylvania
Philadelphia, PA 19104, USA
gee@mail.med.upenn.edu)

Abstract: Mobile cloud computing is referred as the combination of cloud computing and mobile networks to bring benefits for both mobile users and cloud computing providers. While once the data of mobile users is outsourced to the cloud, it is a formidable and challenging task for the data owners to realize both the data confidentiality and the utilization because it seems unachievable to search and retrieve the special contents on the data encrypted by traditional encryption schemes. To address this issue, we propose a searchable public-key encryption scheme for a group of users in mobile cloud storage. In our proposal, a dynamic asymmetric group key agreement protocol is utilized for data sharing among a body of mobile users and the technique of proxy re-signature is employed to update the searchable ciphertexts when the mobile users in the group varies. Through the security proof and performance evaluation, we demonstrate the new scheme is both secure and efficient, and hence it reaches the requirements of the users, network operators, as well as cloud computing providers in application.
Key Words: Cloud storage, mobile network, mobile cloud computing, searchable encryption, data sharing
Category: E.3

1 Introduction

In Big Data era, a huge volume of data has been created recently, it is hard for us to store, share, analyze and utilize the existing data using processing devices [Lu et al. 2014], especially some resource-limited devices, such as smart phones and tablet pcs, which have become an essential part of communication tools without the bound of time and space. Meanwhile, cloud computing is emerged as a new generation of computing infrastructure that offers some appealing advantages and allows users to use the infrastructure, platforms and softwares offered by cloud providers as services at low cost. As a result, with the explosion of mobile applications and a variety of services for mobile users, mobile cloud computing (MCC) [Kumar et al. 2010, Rimal et al. 2009, Canepa et al. 2010] is introduced as an integration of cloud computing into the mobile environment, which brings new types of facilities and services for mobile users to take full advantages of cloud computing [Dinh et al. 2013].

Mobile cloud storage, a dominate type of services offered by mobile cloud service providers, allows mobile users to outsource their data such as contacts, calenders and SMS to the cloud and access them without the restriction on the space and time through the wireless networks. One attractive superiority of mobile cloud storage is that the risk of data loss is significantly reduced since mobile phones are always vulnerable to being dropped, stolen or lost for example. However, even though mobile cloud storage makes these advantages more appealing than ever, it inherits the security threats of conventional cloud computing and causes a group of challenges that are particular to mobile devices offloading jobs through wireless communication channels [Fernando et al. 2013]. Once the files are outsourced to cloud server to extend the storage capacity, mobile users lose the physical control of their data simultaneously. The loss of control can trigger challenging issues that related to confidentiality problem in the cloud. According to the report that released by the Cloud Vulnerabilities Working Group of the cloud security alliance (CSA) [CSA 2011], Data Loss & Leakage is the second threat that just happens less frequently than Insecure Interface & APIs among seven threat types defined by CSA. Gmail's mass email deletion incident [Arrington 2006], Apple's MobileMe's post-launch downtime [Krigsman 2008] and T-Mobile Sidekick users' personal data loss incident [Sidekick 2009] are all such examples. Therefore, the confidentiality protection is an essential problem that should be addressed urgently to avoid data leakage in these incidents.

Even though data encryption is able to prevent the data from being captured by malicious adversaries, the data encrypted by the keys of the cloud servers still would be revealed to the unfully-trusted cloud vendors. In this reason, mobile users have to protect the data using their own keys before upload the data to the cloud, but this mechanism raises a challenging task of data utilization, which means that it is hard to search and retrieve the special contents on the data encrypted by using traditional encryption schemes. To tackle this problem, Song et al. [Song et al. 2000] proposed the notion of searchable encryption and constructed a concrete scheme from symmetric encryption that enables to search on the encrypted data without any loss of data confidentiality. Later, Boneh et al. [Boneh et al. 2004] put forth the concept of public key encryption with keyword search (PEKS) and built a PEKS scheme for providing encrypted email processing capability. Consequently, a various kinds of extensions were presented to adapt to the different scenarios in reality, including conjunctive keyword search [Zhang et al. 2011, Golle et al. 2004], searchable encryption with designed tester [Rhee et al. 2009, Beak et al. 2008], etc. Recently, as the proliferation of cloud computing, the topic that how to effective and privacy-preserving search on the encrypted cloud data becomes a research hotpot and a variety of schemes are proposed to meet the diverse requirements of users, such as fuzzy keyword search [Li et al. 2010, Liu et al. 2011, Wang et al 2014], ranked keyword search [Wang et al. 2010, Cao et al. 2011] and top-k keyword search [Yu et al. 2013]. Unfortunately, these schemes are all designed for traditional cloud storage environment without considering the applications for mobile cloud.

In mobile cloud storage, data sharing among a group of mobile users is one of the most beneficial properties. For example, it is frequent to share some photos and contacts among friends and documents for colleagues. In addition, due to the feature of the mobility, the group members change constantly, including members' joining and leaving. A trivial method to achieve group dynamic operations is to retrieve and decrypt the shared file, then encrypt it using the new key that shared among the new member in the group and upload the encrypted data to the cloud. This approach is inefficient due to the heavy computational and communication costs. Therefore, a searchable encryption scheme for mobile cloud storage is supposed to support data sharing as well as group dynamic operations.

In this paper, we propose a public key encryption with keyword search scheme that supporting data sharing among multiple mobile users in a dynamic group. As far as we know, our work is among the first few ones to achieve the privacy-preserving keyword search on encrypted data in mobile cloud storage. Our contributions can be summarized as follows:

1. We motivate the searchable public-key encryption with data sharing for dynamic groups in mobile cloud storage and describe the system model and security threats.

2. Deriving from the group key agreement protocol and proxy re-encryption, we propose a searchable encryption scheme which provides data sharing, group dynamic and efficient ciphertexts updating.

3. We prove the security and justify the performance of our scheme by analyzing the computation, communication and storage overhead.

The rest of the paper is organised as follows. In section 2, we define the system and security models. Then, we describe the scheme in section 3 and provide the security proof for the proposed scheme in section 4, respectively. We analyze the performance of our proposal in section 5 and conclude the paper in section 6.

2 Problem statement

In this section, we describe both the system model and the security model of the searchable encryption scheme in mobile cloud storage environment.

2.1 The system model

Mobile cloud storage service consists of two parties: mobile users and cloud servers. Mobile users can access the wireless networks using some mobile devices and have a large number of data files to store while their storage space is limited. Cloud servers are managed by the cloud server vendors and provides cloud storage service to mobile users relying on their significant storage space and computation resources.

Every party has its own obligations and benefits. Mobile users enjoy the convenience to store a multitude of files in cloud and share them among group members. Upon a cloud user in the group is corrupted because of say, economic interests, he will be revoked and got rid of the group by other group users. When some mobile user tries to search the data uploaded to the cloud by other users before, he generates a trapdoor from the required keyword and forwards it to the server. The cloud server will perform the test task honestly to check whether the target keyword is contained in the encrypted data without exposing any contexts of the data.

2.2 System components

A searchable public-key encryption scheme with data sharing consists of seven algorithms: KeyGen, GkeyGen, PEKS, Trapdoor, Test, Join and Leave as follows.

1. KeyGen: Taking a security parameter κ as inputs, this algorithm computes a public-secret key pair (pk_i, sk_i) for each mobile user U_i in the group $S = \{U_1, \cdots, U_n\}$, where n is the number of mobile users and makes pk_i public.
2. GkeyGen: Taking the security parameter κ and every (pk_i, sk_i) in S as inputs, this algorithm outputs a group public-secret key pair (Gpk, Gsk), and releases Gpk.
3. PEKS: Taking the public parameter κ, the group public key Gpk and a set of the selected keyword $W = \{w_1, \cdots, w_s\}$ as inputs, this algorithm generates the searchable ciphertexts C_i for each w_i.
4. Trapdoor: Taking the public parameter κ, the group secret key Gsk and a chosen keyword w', this algorithm outputs a trapdoor $T_{w'}$ for w'.

5. Test: Taking the public parameter κ, the group public key Gpk, the searchable ciphertexts C_i and the trapdoor $T_{w'}$ as inputs, the algorithm returns the corresponding data if w' is one of keywords in W; Otherwise, returns $\perp$.

6. Join: Taking each (pk_i, sk_i) in new group $S' = \{U_1, \cdots, U_n, U_{n+1}, \cdots, U_{n+n'}\}$ as inputs, where $\{U_{n+1}, \cdots, U_{n+n'}\}$ are the new joined members, this algorithm generates a new group public-secret key pair (Gpk', Gsk'), and updates the searchable ciphertexts.

7. Leave: Taking each (pk_i, sk_i) in new group $S' = \{U_1, \cdots, U_{i-1}, U_{i+1}, \cdots, U_n\}$ as inputs, where U_i has departed from the group, this algorithm computes a new group public-secret key pair (Gpk', Gsk'), and updates the searchable ciphertexts.

2.3 Security model

The security of searchable public-key encryption schemes follows the property of indistinguishability of searchable ciphertexts against a chosen keywords attack (IND-CKA) due to Boneh et al. [Boneh et al. 2004]. In order to prevent the adversary from obtaining the capacity of generating a valuable group secret key from GkeyGen, Join and Leave phases, we extend the model by adding the GkeyGen, Join and Leave queries. The new security game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ is shown as follows:

- KeyGen: The challenger $\mathcal{C}$ runs KeyGen algorithm to generate a series of public-secret key pairs (pk_i, sk_i) for mobile users. It sends the public keys pk_i to the adversary $\mathcal{A}$ and keeps sk_i secret.

- Queries 1: $\mathcal{C}$ responses the queries launched by $\mathcal{A}$ adaptively.

  1. GkeyGen queries: $\mathcal{A}$ chooses a group $S = \{U_1, \cdots, U_n\}$ to query adaptively. $\mathcal{C}$ generates the group public-secret key pair (Gpk, Gsk) for the group S and responses it to $\mathcal{A}$.

  2. Trapdoor queries: $\mathcal{A}$ can query the trapdoor of any chosen keyword w' and a user in group S. $\mathcal{C}$ runs the Trapdoor algorithm and returns the trapdoor to $\mathcal{A}$.

  3. Join queries: $\mathcal{A}$ chooses the group key pair under the group S that received from GkeyGen queries and some new users $S' = \{U'_1, \cdots, U'_{n'}\}$ to query adaptively. $\mathcal{C}$ generates the group public-secret key pair (Gpk, Gsk) for the group $S" = S + S'$, and then responses them to $\mathcal{A}$.

  4. Leave queries: $\mathcal{A}$ chooses the group key pair under the group S that received from GkeyGen queries and a leaved user U_i to query adaptively.

$\mathcal{C}$ generates the group public-secret key pair (Gpk, Gsk) for the group $S' = S \setminus U_i$, and then responses them to $\mathcal{A}$.

– Challenge: $\mathcal{A}$ selects two target keywords (w_0, w_1) with a group S^*, and forwards them to $\mathcal{C}$. The restriction here is that w_0 and w_1 should not be issued in Trapdoor queries 1, S^* should never be asked in GkeyGen queries 1, the subset of S^* should not be queried in Join queries 1 and S^* is not the subset of the collection asked in Leave queries 1. Upon receiving (w_0, S^*) and (w_1, S^*), $\mathcal{C}$ picks a random $\beta \in \{0, 1\}$, and computes the searchable ciphertext C_β for w_β, then returns C_β to $\mathcal{A}$.

– Queries 2: $\mathcal{C}$ answers the GkeyGen, trapdoor, Join and Leave queries as in queries 1. The restriction here is that w_0 and w_1 should not be issued in Trapdoor queries, S^* should never be asked in GkeyGen queries, the subset of S^* should not be queried in Join queries and S^* is not the subset of the collection asked in Leave queries.

– Guess: Finally, $\mathcal{A}$ outputs its guess $\beta' \in \{0, 1\}$ and wins the game if $\beta' = \beta$.

The advantage of $\mathcal{A}$ is defined as $\text{Adv}^{IND-CKA}(\mathcal{A}) =\mid \Pr[\beta^{'} = \beta] - \frac{1}{2} \mid$. The PERKS scheme is said to be (τ, ε)-IND-CKA secure if for any $\mathcal{A}$, the guessing advantage $\text{Adv}^{IND-CKA}(\mathcal{A})$ is less than ε in polynomial time τ.

3 Our construction

Our searchable encryption with group dynamic protocol derives from public-key encryption with keyword search scheme due to Boneh et al. For the join and revocation of mobile users, we resort to the dynamic asymmetric group key agreement scheme [Zhao et al. 2011], where all users in a temporary group negotiate a public-secret key pair. In order to update the searchable ciphertexts, we utilize the idea of proxy re-encryption, which enables a semi-trusted proxy to transform ciphertexts on m that can be decrypted by Alice into Bob's ciphertexts on m. The details of the protocol are as follows. Let q be a large prime and G and G_T be two multiplicative cyclic groups with the same prime order p, and g be a generator of G. $\hat{e} : G \times G \to G_T$ denotes a bilinear map and $H : \{0, 1\}^* \to Z_p^*$, $H_1 : \{0, 1\}^* \to G$, $H_2 : Z_p^* \to G$ and $H_3 : G_T \to G$ represent four cryptographic hash functions. We assume that n mobile users form a temporary group $S = \{U_1, \cdots, U_n\}$ to share a file m.

1. KeyGen: U_i chooses a random value $x_i \in Z_p^*$ as its secret key sk_i and computes a public key $pk_i = g^{x_i}$. The public key g^{x_i} is released and the secret key x_i is kept privately.

2. GkeyGen: A group of mobile users $S = \{U_1, \cdots, U_n\}$ form a circle structure, with $U_{n+1} = U_1$, $U_0 = U_n$, to negotiate to share a public-secret key pair (Gpk, Gsk). The system time N is used to denote the unique identifier of this group. (Gpk, Gsk) are generated in the following steps:

   – Step 1: For every U_i, it firstly calculates a shared key with neighbours $pk_{i,i+1} = pk_{i+1}^{x_i}$ and $pk_{i-1,i} = pk_{i-1}^{x_i}$. Then U_i computes $X_i = H(pk_{i,i+1}) \oplus H(pk_{i-1,i})$ and $M_i = U_i \mid\mid N \mid\mid H(S) \mid\mid X_i$. At last, it broadcasts M_i to other mobile users in group S.

   – Step 2: Upon receiving M_i from all users, Each U_i checks whether $X_1 \oplus \cdots \oplus X_n \stackrel{?}{=} 0$. If it is valid, U_i rejects by emitting 0 and aborts; Otherwise computes a group secret key Gsk as:

   $$Gsk = H(H(pk_{1,2}) \mid\mid \cdots \mid\mid H(pk_{i,i+1}) \mid\mid \cdots \mid\mid H(pk_{n,1}) \mid\mid N),$$

   where

   $$H(pk_{i-j,i-j+1}) = H(pk_{i,i-1}) \oplus X_{i-1} \oplus \cdots \oplus X_{i-j}.$$

   for each $j = \{1, \cdots, n-1\}$.

   – Step 3: Every U_i in group S computes a group public key $Gpk = g^{Gsk}$ and broadcasts it to other members.

3. PEKS. When some user U_i tries to share a file m with other members in group $S = \{U_1, \cdots, U_n\}$, it picks a collection of keywords $W = \{w_1, \cdots, w_s\}$, and performs as follows:

   – U_i firstly picks a key pair (ssk, svk) for one-time signature scheme and s random values $b_1, \cdots, b_s \in Z_p^*$. Then, he computes $W_0 = \hat{e}(g, H_2(svk))$ and $W_i = \hat{e}(g, H_1(w_i))$ for each keyword $w_i, 1 \leq i \leq s$.

   – U_i randomly picks $r \in Z_p^*$ and calculates the ciphertext of m: $B = Gpk^r$, $C = m \cdot W_0^r$, $D = H_2(svk)^r$.

   – To compute the searchable ciphertext C_i for each w_i, U_i picks a random $r_i \in Z_p^*$ and calculates

   $$C_i = [C_{i1}, C_{i2}] = [Gpk^{r_i}, H_3(t_i)] = [Gpk^{r_i}, H_3(W_i^{r_i})]. \tag{1}$$

   – Finally, U_i generates the one-time signature $\sigma = S_{ssk}(C, D, C_{12}, \cdots, C_{s2})$ and stores $\{svk, B, C, D, C_1, \cdots, C_s, \sigma\}$ to the cloud server.

4. Trapdoor. When some U_j in the group wants to search the shared data, he uses the group secret key Gsk and the queried keyword w' to compute the corresponding trapdoor $T_{w'} = H_1(w')^{-Gsk} \in G_1$.

5. Test. Upon receiving the trapdoor from U_j, the server checks whether the following equation holds for every $C_i = (C_{i1}, C_{i2})$:

$$H_3(\hat{e}(T_{w'}, C_{i1})) \stackrel{?}{=} C_{i2}. \tag{2}$$

   If it is valid, the server returns m's ciphertext (B, C, D) with (svk, σ) to U_j; Otherwise, outputs $\perp$. When U_j receives the response, he verifies the availability of σ using svk. If σ is also valid, U_j decrypts the ciphertext as $m = C/\hat{e}(B, H_2(svk))^{-Gsk}$; Otherwise, outputs $\perp$.

6. Join: We suppose certain outsiders $J = \{U_{n+1}, \cdots, U_{n+n'}\}$ hope to join the current group S. They form a new circle structure $S' = \{U_1, \cdots, U_{n+n'}\}$ with $U_{n+n'+1} = U_1$, $U_0 = U_{n+n'}$. A new group identifier N' is chosen from the system time. The mobile users calculate a new group public-secret key pair (Gpk', Gsk') and update the tags as follows:

   – Step 1: U_1, U_n and $J = \{U_{n+1}, \cdots, U_{n+n'}\}$ follows the step 1 in GkeyGen phase to broadcast M_i. $pk_{1,2}$ and $pk_{n-1,n}$ are unchanged and the remaining users $\{U_2, \cdots, U_{n-1}\}$ re-publish the previous M_i.

   – Step 2: All mobile users in S' generate a group secret key Gsk' following Step 2 in GkeyGen phase and calculate the corresponding group public key $Gpk' = g^{Gsk'}$.

   – Step 3: Some user in S uses Gsk and Gsk' to calculate a proxy re-encryption key $ReGsk = Gsk'/Gsk$.

   – Step 4: Upon receiving $ReGsk$, the server computes $B^* = B^{ReGsk}$ and $C^*_{i1} = C_{i1}^{ReGsk}$ for each C_i that related to w_i, and updates the data in cloud.

7. Leave: We assume that U_i leaves the group S and the remainders form a new circle structure among users $S' = \{U_1, \cdots, U_{i-1}, U_{i+1}, \cdots, U_n\}$. A new group identifier N' is chosen based on system time. The remaining mobile users compute a new group public-secret key pair (Gpk', Gsk') and update the tags as follows:

   – Step 1: U_{i-1} and U_{i+1} follows the step 1 in GkeyGen phase to broadcast M_i. $pk_{n-2,n-1}$ and $pk_{n+1,n+2}$ are unchanged and the rest users re-publish the previous M_i.

   – Step 2: All mobile users in S' calculate the group secret key Gsk' according to Step 2 in GkeyGen algorithm and generate the corresponding group public key $Gpk' = g^{Gsk'}$.

– Step 3: Some user in S uses Gsk and Gsk' to compute a proxy re-encryption key $ReGsk = Gsk'/Gsk$.

– Step 4: Upon receiving $ReGsk$, the server computes $B^* = B^{ReGsk}$ and $C^*_{i1} = C^{ReGsk}_{i1}$ for each C_i that related to w_i, and updates the data in cloud.

3.1 Correctness

The group key agreement scheme can ensure that all users in group S can obtain the same group secret key Gsk after communicating with other members. The ciphertexts of m can be decrypted as follows,

$$\begin{aligned} m &= C/\hat{e}(B, H_2(svk))^{-Gsk} \\ &= m \cdot W_0^r/\hat{e}(B, H_2(svk))^{-Gsk} \\ &= m \cdot \hat{e}(g, H_2(svk))^r/\hat{e}(Gpk^r, H_2(svk))^{-Gsk} \\ &= m \cdot \hat{e}(g, H_2(svk))^r/\hat{e}(g, H_2(svk))^r \\ &= m. \end{aligned}$$

The consistency of searchable encryption holds because

$$\begin{aligned} C_{i2} &= H_3(W_i^{r_i}) \\ &= H_3(\hat{e}(g, H_1(w_i)^{r_i}) \\ &= H_3(\hat{e}(T_{w'}, C_{i1})). \end{aligned}$$

In proxy re-encryption, the cloud server transforms the ciphertext on m and the searchable ciphertexts on w_i that are only decrypted by the previous group members into the ciphertexts that the members in an updated group can decrypt. In the ciphertexts on m, (B, C, D), just the element B is generated using the group public key Gpk, and thus the correctness of the proxy re-encryption on m can be shows as:

$$B^* = B^{ReGsk} = B^{Gsk'/Gsk} = g^{rGsk \cdot Gsk'/Gsk} = g^{rGsk'} = Gpk'^r. \tag{3}$$

In the searchable encryption, the ciphertexts $C_i = [Gpk^{r_i}, H_3(T_i^{r_i \cdot b_i^{-1}})]$, in which C_{i1} is concerned with Gsk, so C_{i1} is shifted in a correctness way:

$$C^*_{i1} = C^{ReGsk}_{i1} = Gpk^{r_i \cdot Gsk'/Gsk} = g^{r_i Gsk \cdot Gsk'/Gsk} = g^{r_i Gsk'} = Gpk'^{r_i}. \tag{4}$$

4 Security Proof

Theorem 1. The group public-secret key pair is securely computed as long as all mobile users in group are honest.

Proof. The security of the group key generation can be shown as there is no adversary that can get enough information to generate a valid group secret key. This proof is straight-forward. Our method of generating group public-secret key pair derives from the dynamic asymmetric group key agreement scheme [Zhao et al. 2011]. If the mobile users in group are honest, the correctness of the group key agreement scheme ensures to generate a shared group secret key. According to the security proof in [Wu et al. 2008], the group public-secret key pair is secure if Diffie-Hellman key agreement scheme is secure, whose security can be reduced to CDH assumption.

Theorem 2. The group secret key is not disclosed with respect to the joining or leaving mobile users as long as s-CDH assumption and s-CDHI assumption hold.

Proof. In GkeyGen phase, Diffie-Hellman key agreement scheme is reused for $n+1$ times to compute the group secret keys for dynamic groups. Actually, $g^x, g^{x^2}, \cdots, g^{x^{2^v}}$ are immediate values used for generating the group secret keys. We firstly consider the joining case. Suppose a mobile user joins the group in the $(i+1)$th key exchange process, the joining user knows $g^{x^{2^i+1}}$ possibly along with some subsequent items. Here we consider the extreme case in which he knows $Q, Q^x, Q^{x+1}, \cdots, Q^{x^w}(Q = g^{x^{2^i+1}}, w = v-2^i-1, 0 \leq i < v)$ and tries to compute the ith group secret key. If s-CDHI assumption holds, the joining mobile user is unable to compute $Q^{1/x} = g^{x^{2^i}}$. In addition, since the target group secret key is computed using a hash function, the joining member can not retrieve any information about it. Thus, the group secret key is secure with respect to joining members.

Regarding the leaving case, we assume a mobile user leaves the group in the ith key exchange process. The leaving mobile user could know $g^{x^{2^{i-1}}}$ along with some foregoing items. Here we consider an extreme case in which he knows $Q, Q^x, Q^{x+1}, \cdots, Q^{x^s}(Q = g^x, w = 2^i-2, 0 < i \leq v)$ and tries to generate the ith group secret key. If s-CDH assumption holds, the leaving mobile user is unable to compute $Q^{x^{w+1}} = g^{x^{2^i}}$. Besides, since the target group secret key is computed via a hash function, the leaving member can not retrieve any information about it. Thus, the group secret key is secure with respect to the leaving members.

Theorem 3. The encryption of the file m is CCA secure in the random oracle model if the DBDH assumption holds.

Proof. In the PEKS phase, the mobile user encrypts the file m to generate the ciphertexts that can be updated to new ciphertexts when the members in the group varies. In the algorithms of Join and Leave, an honest mobile user acts as a proxy to re-encrypt the ciphertexts of m using the proxy re-encryption key. In order to distinguish the ciphertexts returned from the challenger, the adversary can make the Leave queries and Join queries. The capability of the attacker that can get from these queries is the same as that of the attacker who strive to break the proxy re-encryption scheme proposed by Canetti and Hohenberger [Canetti et al. 2007]. As a consequence, the security of the encryption can be reduced to the DBDH assumption which the underlying proxy re-encryption scheme depends on.

Theorem 4. The adversary can not distinguish the searchable ciphertexts even though it can make trapdoor queries, Join queries and Leave queries, if the mBDH problem is computationally hard.

Proof. The searchable ciphertexts are generated based on the PEKS scheme due to Boneh et al. [Boneh et al. 2004] and a proxy re-encryption key is used to transform the old ciphertexts to the new ones that only can be searched by the new members of a updated group. By employing the Join queries and Leave queries, the adversary can obtain some proxy re-encryption keys and it can get some trapdoors of chosen keywords from trapdoor queries. The probable security threats have been captured by the security model of proxy re-encryption with keyword search proposed by Yau et al. [Yau et al. 2011]. Following the security proof of the proxy re-encryption with keyword search scheme, it is easy to tell that the distinguishability of the searchable ciphertexts in our construction relies on the security of proxy re-encryption with keyword search scheme, whose distinguishability can be reduced to the mBDH assumption.

5 Performance Analysis

Here we will demonstrate the efficiency analysis of our scheme. By efficient we mean that the proposed scheme provides the desired function of searching on encrypted data among multiple users while incurring minimal computation, communication and storage overhead. We mainly focus the computation, communication and storage burden incurred by our new protocol and report its efficiency for dynamic groups.

Communication Cost. In GkeyGen phase, every mobile user broadcasts M_i to other group users which is of binary length 4 log_2p. In the PEKS phase, some user U_i generates the searchable ciphertexts for the chosen keywords $W = \{w_1, \cdots, w_s\}$ and the ciphertext for the file m, and then forwards them to the cloud server. The message $\{svk, B, C, D, C_1, \cdots, C_s, \sigma_m\}$ has the length of $(2s+6)log_2q + log_2p$ bits, including the size of one-time signature. When one of the mobile users U_j is willing to retrieve the data, he computes the trapdoor $T_{w'}$ which is only of binary length log_2q according to the keyword w' which he wants to search. Upon receiving the trapdoor, the server determines whether the queried keyword is one of the elements in W and returns the results to the users. The results is "$\perp$" that is only one bit or $log_2p + (s+6)log_2q$-bit data. In Join and Leave phase, the mobile user should send the proxy re-encryption key $ReGsk$ to update the ciphertexts, thus, the communication cost is log_2q bits.

Storage Cost. In terms of the storage cost, both the data and the searchable ciphertexts are held at the server side. So the mobile users only need to maintain his own and group public-secret key pairs, which cost $2\ log_2p+2\ log_2q$ bits. Apart from these, the mobile users have to store M_i in case of updating, which is $4\ log_2p$ bits. As for the cloud server, he contributes to store and manage both the data and searchable ciphertexts of the binary length $(2s+6)log_2q + log_2p$.

Computation Cost. To evaluate computation overhead on the mobile users and server, we specify P, Exp, Mul_p, Mul_q to denote the pairing computation, the exponentiation in Z_p^*, the multiplication in Z_p^*, the multiplication in G respectively. Table 1 summarizes analytical result of each entity's computation overhead on every algorithm. On the side of the users, we utilize the average value of the computation overhead to indicate the efficiency.

Table 1: Computation analysis

Component	Overhead
KeyGen	$1Exp$
GkeyGen	$3Exp$
PEKS	$(s+1)P + (2s+3)Exp + Mul_q$
Trapdoor	$1Exp$
Test	$P + Exp + Mul_q$
Join	$\frac{(2n'+n+2)Exp+Mul_p}{n+n'}$
Leave	$\frac{(2+n')Exp+Mul_p}{n'}$

Note that it might be still a bit hard for the mobile devices to perform the bilinear pairing computation even though their capacity has been improved significantly in recent years. So we show two approaches to solve this problem Sophisticatedly. The first method is the technique of pre-computation. Before the mobile user wants to outsource the data, it can compute the bilinear pairing in the suitable devices to avoid the heavy burden of computing on mobile devices. The second way is to utilize the computation outsourcing to outsource the bilinear pairing computation to the server which has significant computation resources to aid the users.

6 Conclusions

In this paper, we focus on the encrypted data search and retrieval problem for mobile cloud storage and propose a searchable public-key encryption with data sharing for dynamic groups. We utilize the dynamic group agreement scheme to guarantee that every mobile user in groups can share the same group secret key and update it when the members of groups varies. Considering that the ciphertexts that should be able to be decrypted by the new members in the group, the technique of proxy re-encryption is employed to address the ciphertexts updating issue. Through the detailed security proof and performance analysis, we demonstrate our proposed scheme is provable secure and efficient to be implemented in the mobile cloud storage scenario. For the future work, we will study the privacy-preserving keyword search for shared data and extend our proposed scheme to effectively resist the off-line keyword guessing attack.

Acknowledge This work is supported in part by the Postdoctoral Science Foundation of China(2013M542267), Open Research Foundation of Integrated Electronic System of the Ministry of Education of China(20120105), and the Fundamental Research Funds for the Central Universities(ZYGX2013J118, ZYGX2013J079).

References

[Arrington 2006] Arrington, M.: "Gmail disaster: reports of mass e-mail deletions" (2006) `http://www.techcrunch.com/2006/12/28/gmail-disaster-reports-of-massemail-deletions/index.html`.

[Beak et al. 2008] Beak, J., Safavi-Naini, R., Susilo, W.: "Public key encryption with keywod search revisited"; Proc. of ICCSA 08, LNCS 5072, 2008, 1249-1259.

[Boneh et al. 2004] Boneh, D., Crescenzo, G. D., Ostrovsky, R., Persiano, G.: "Public key encryption with keyword search"; Proc. Advances in Cryptology-EUROCRYPT 2004 Lecture Notes in Computer Science, Vol. 3027, Cachin C, Cameinisch M(eds.). springer, 2004, 506-522.

[Canepa et al. 2010] Canepa, H., Lee, D.: "A virtual cloud computing provider for mobile devices"; Proc. 1^{st} ACM Workshop on Mobile Cloud Computing and Services Social Networks and Beyond (MCS 2010), San Francisco, USA, no. 6 ACM Digital Library (2010), 6.

[Canetti et al. 2007] Canetti, R., Hohenberger, S.: "Chosen-ciphertext secure proxy re-encryption"; Proc. of ACM CCS 2007, ACM New York, NY, USA, 2007, 185-194.

[Cao et al. 2011] Cao, N., Wang, C., Li, M., Ren, K., Lou, W. J.: "Privacy-preserving multi-keyword ranked search over encrypted cloud data"; Proc. of IEEE INFOCOM'11 Conference, Shanghai, China, 2011.

[CSA 2011] Cloud Vulnerabilities Working Group of the cloud security alliance,"Cloud Computing Vulnerability Incidents: A Statistical Overview" (2011) `https://cloudsecurityalliance.org/research/vulnerabilities/#_downloads`.

[Dinh et al. 2013] Dinh, H.T., Lee, C., Niyato, D., Wang, P.: "A survey of mobile cloud computing: architecture, applications, and approaches"; Wireless Communication and Mobile Computing, 13, 8 (2013) 1587-1611.

[Fernando et al. 2013] Fernando, N., Loke, S.W., Rahayu, W.: "Mobile cloud computing: a survey"; Future Generation Computer Systems, 29 (2013) 84-106.

[Golle et al. 2004] Golle, P., Staddon, J., Waters, B.: "Secure conjunctive search over encrypted data"; Proc. Applied Cryptography and Network Security-ACNS'04, Lecture Notes in Computer Science, Vol.3089, Jakobsson M, Yung M, zhou JY(eds.). Springer Berlin/Heidelberg, 2004, 31-45.

[Huang et al. 2013] Huang, D., Xing, T., Wu, H.: "Mobile cloud computing service models: a user-centric approach"; IEEE Network, 27, 5 (2013), 6-11.

[Krigsman 2008] Krigsman, M.: "Apple's mobileme experiences post-launch pain" (2008) http://blogs.zdnet.com/projectfailures/?p=908.

[Kumar et al. 2010] Kumar, K., Lu, Y.H.: "Cloud computing for mobile users: can offloading computation save energy?"; IEEE Journal Computer, 43, 4 (2010) 51-56.

[Li et al. 2010] Li, J., Wang, Q., Wang, C., Cao, N., Ren, K., Lou, W. J.: "Fuzzy keyword search over encrypted data In cloud computing"; Proc. of IEEE INFOCOM'10 Mini-Conference, San Diego, CA, USA, 2010.

[Liu et al. 2011] Liu, C., Zhu, L., Li, L., Tan, T.: "Full keyword search on encrypted cloud storage data with Small Index"; Proc. 2011 IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS), Beijing, China, 2011.

[Lu et al. 2014] Lu, R., Zhu, H., Liu, X., Liu, J.K., Shao, J.: "Toward efficient and privacy-preserving computing in Big Data era"; IEEE Network, 28, 4 (2014), 46-50.

[Rhee et al. 2009] Rhee, H. S., Park, J. H., Susilo, W., Lee, D. H.: "Improved searchable public key encryption with designated tester"; Proc. 4^{th} International Symposium on Information, Computer, and Communications Security-ASIACCS'09, 2009, 376-379.

[Rimal et al. 2009] Rimal, B.P., Choi, E., Lumb, I.: "A taxonomy and survey of cloud computing systems"; Proc. 5^{th} International Joint Conference of INC, IMS and IDC, NCM 2009, Seoul, Korea, IEEE Press (2009), 44-51.

[Sidekick 2009] Shiels, M.: "Phone sales hit by sidekick loss" (2009) http://news.bbc.co.uk/2/hi/technology/8303952.stml

[Song et al. 2000] Song, D., Wagner, D., Perrig, A.: "Practical techniques for searching on encrypted data"; Proc. 2000 IEEE symposium on research in security and privacy, Berkeley, California, USA, (2000).

[Wang et al. 2010] Wang, C., Cao, N., Li, J., Ren, K., Lou, W.J.: "Secure ranked keyword search over encrypted cloud data"; Proc. of ICDCS'10, Genoa, Italy, 2010.

[Wang et al 2014] Wang, B., Yu, S., Lou, W., Hou, Y. T.: "Privacy-Preserving Multi-Keyword Fuzzy Search over Encrypted Data in the Cloud"; IEEE INFOCOM 2014, Toronto, Canada, 2014.

[Wu et al. 2008] Wu, S., Zhu, Y.: "Constant-round password-based authenticated key exchange protocol for dynamic groups"; Proc. of FC 2008, LNCS 5143, 2008, 69-82.

[Yau et al. 2011] Yau, W.-C., Phan, R. C.-W., Heng, S.-H., Goi, B.-M.: "Proxy Re-encryption with Keyword Search: New Definitions and Algorithms with Proofs"; International Journal of Security and Its Applications, 5, 2 (April 2011) 75-90.

[Yu et al. 2013] Yu, J., Liu, P., Zhu, Y., Xue, G., Li, M.: "Towards secure multi-keyword top-k retrieval over encrypted cloud data"; IEEE Transactions on Dependable and Secure Computing, 10, 4 (2013) 239-250.

[Zhang et al. 2011] Zhang, B., Zhang, F.: "An efficient public key encryption with conjunctive-subset keywords search"; Journal of Network and Computer Applcations, 34, 1 (2011) 262-267.

[Zhao et al. 2011] Zhao, X., Zhang, F., Tian, H.: "Dynamic asymmetric group key agreement for ad hoc networks"; Ad Hoc Networks, 9 (2011) 928-939.