PLOS ONE

RESEARCH ARTICLE

Searchable attribute-based encryption scheme with attribute revocation in cloud storage

Shangping Wang[1], Duqiao Zhao[1]*, Yaling Zhang[2]

1 School of Science, Xi'an University of Technology, Xi'an, Shaanxi, China, **2** School of Computer Science, Xi'an University of Technology, Xi'an, Shaanxi, China

* zduqiao@163.com

Citation: Wang S, Zhao D, Zhang Y (2017) Searchable attribute-based encryption scheme with attribute revocation in cloud storage. PLoS ONE 12 (8): e0183459. https://doi.org/10.1371/journal.pone.0183459

Editor: Yeng-Tseng Wang, Kaohsiung Medical University, TAIWAN

Received: October 19, 2016

Accepted: August 6, 2017

Published: August 31, 2017

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Funding: This work is supported by the National Natural Science Foundation of China under grants 61572019, 61173192, and the Key Project of Research Foundation of Natural Science Foundation of Shaanxi Province of China under Grant No. 2016JZ001.

Competing interests: The authors have declared that no competing interests exist.

Abstract

Attribute based encryption (ABE) is a good way to achieve flexible and secure access control to data, and attribute revocation is the extension of the attribute-based encryption, and the keyword search is an indispensable part for cloud storage. The combination of both has an important application in the cloud storage. In this paper, we construct a searchable attribute-based encryption scheme with attribute revocation in cloud storage, the keyword search in our scheme is attribute based with access control, when the search succeeds, the cloud server returns the corresponding cipher text to user and the user can decrypt the cipher text definitely. Besides, our scheme supports multiple keywords search, which makes the scheme more practical. Under the assumption of decisional bilinear Diffie-Hellman exponent (q-BDHE) and decisional Diffie-Hellman (DDH) in the selective security model, we prove that our scheme is secure.

Introduction

In 2005, Waters et al.[1] came up with the concept of ABE(Attribute-Based Encryption) which was much more flexible than traditional public-key encryption. With the development and deepening of ABE, the attribute revocation of ABE is concerned by more and more people. The efficient attributes revocation scheme is an integral part of ABE scheme, which is one of the difficulties for the application of ABE, and the study of ABE is inseparable from the attribute revocation scheme research.

P. Traynor et al.[2] put forward a scheme which achieved the update of secret key in 2006. However, it needed that the user must kept close contact with attribute authority to get the secret key. Thereafter, Kumar et al.[3] presented a scheme with revocation of ABE, and it expanded from the IBE which they proposed before. All of these articles demand that users need to access the attribute authority for key reissuing at regular intervals.

In 2008, Jiang et al.[4] gave a scheme that solved the key misused problem of users. However, in this scheme, the third party should be included in each decryption key of users, and made it was unrealistic. After that, Kim et al.[5] inserted the users' information in the secret

key of attribute by using the black box model and sent it to the user, which was more efficient to guarantee the security of the system.

Attrapadung et al.[6] put forward the two revocation models, they are direct revocation model and indirect revocation model. The direct revocation model is specified the revocation list by sender, and the indirect revocation model updates the secret key periodically by the key center. In [7] [8], the authors gave some ABE instances. However, in the above schemes, they do not relate to the keyword search issue, which makes users can not effectively search for files.

To overcome this problem, Boneth et al. [9] proposed a single keyword search scheme, namely the user can only search a single keyword. In this scheme, the data owner extracted the keywords from the file before encrypted, and used the public key to encrypt the keywords. After that, the data owner sent the file and the index of the keywords to the cloud server. The user could generate the search token about the keywords which he wanted to search and sent it to the cloud server. The cloud server used the matching algorithm to find out the cipher text and returned it if the match was successful.

Searchable encryption has many practical applications. In 2011, Kerschbaum et al.[10] proposed a secure conjunctive keyword searches for unstructured text scheme, and the scheme was proved secure in the random oracle model. At the same year, Cao et al.[11] and Chuanh et al.[12] gave schemes that the multi-keyword search over encrypted data.

In 2014, Han et al. [13] proposed an attribute based encryption (ABE) searchable scheme, in which used the homomorphic encryption technology. Sahai et al. [14] gave a outsourcing technique based on the scheme of Gentry et al.[15]. After that, Liang K et al. [16] proposed a searchable ABE mechanism with efficient and secure in cloud storage. This model can be applied to real life, such as the safety of electric power system. And the scheme is secure in the random oracle model. Later, Li et al. [17] proposed a searchable ABE scheme with attribute revocation in cloud storage.

Willy Susilo et al.[18] proposed a searchable scheme, and it supported multiple keywords search. At the same time, Li J et al.[19] made a searchable CP-ABE with revocation. In this scheme, the receivers could not steal any information from the cipher because of the access structures were partially hidden, which made the scheme more secure.

In 2016, Wen et al. [20] proposed a verifiable attribute-based keyword search scheme with fine-grained owner-enforced search authorization in the cloud. This scheme supports user revocation. Besides, it allows data owners encrypt the data and outsource to the cloud server. In the same year, Yang et al. [21] proposed a conjunctive keyword search scheme with designated tester. User can search within a specified time if he is authorized, and it is proved secure in the standard model. In 2017, Jiang et al. [22] proposed a keyword search scheme with efficiency and verification in cloud data, and it allows multi-keyword search. Finally, they gave the security analysis in the scheme. Later, Poon et al.[23] constructed a conjunctive keyword search scheme. This scheme allows phrase search, and has smaller storage cost.

Our contribution

In 2012, Qiang Li et al.[24] put forward a scheme with fine-grained attribute revocation. However, the scheme only achieves the attribute revocation, the keyword search is not involved, this problem may lead to the problem that system users cannot effectively download cipher text which they interested from the cloud server.

In this paper, we propose a keyword search attribute based encryption scheme with attribute revocation. The new scheme supports not only the attribute revocation but also keyword search. When a user wants to search the file which he interests, he sends the search token to

the cloud server, and the cloud server runs the test algorithm. If the test is successful, it returns the file. In this way, the user can download the file which he interests and save the storage space at the same time. Finally, under the assumption of q-BDHE and DDH in the selective security model, we prove that our scheme is secure.

Preliminaries

A linear secret sharing scheme can be used to represent an access control policy (M, ρ), which M is an $l \times k$ matrix, and $S = \{att_1, \ldots, att_n\}$ be an attribute set, and for $i \in [1,l]$, $\rho(i) \rightarrow S$ is a mapping function, and $\rho(i)$ maps a row into the attribute.

Linear Secret-Sharing Scheme (LSSS) [25]

A linear secret sharing scheme includes two algorithms:

Share: In this step, it is dispersing the secret value s to attributes specified by ρ as follows: by selecting $v_2, \ldots, v_k \xrightarrow{R} Z_p$,setting $\vec{V} = (s, v_2, \ldots, v_k)$ and computing $\lambda_i = M_i \cdot \vec{V}$ where M_i is the ith row of M,it assigns secrets share λ_i to the attribute $\rho(i)$.

Combine: In this step, it is used to collect the secret value from secret shares which related to the attributes as follows: selecting subset $I = \{i: \rho(i) \in S\}$ the attribute set $\{\rho(i) \mid i \in I\}$ satisfies access control strategy (M, ρ), and computing coefficients k_i, $i \in I$ such that $\sum_{i \in I} k_i M_i = (1,0,\ldots, 0)$, then we will obtain that $\sum_{i \in I} k_i \lambda_i = s$.

Decisional q-BDHE assumption [24]

The definition of the decisional q-BDHE exponent assumption in our article as follows:

Choose a group G_1 of prime order p, let g be a generator of G_1, and define e: $G_1 \times G_1 \rightarrow G_2$, the adversary is given a vector

$$(g, g^s, g^a, g^{a^2}, \ldots, g^{a^q}, g^{a^{q+2}}, \ldots, g^{a^{2q}}) \in G_1^{2q+1}$$

We say that the Decision q-BDHE assumption holds in G_1 if no polynomial-time algorithm has a non-negligible advantage to distinguish $e(g,g)^{sa^{q+1}}$ and a random element in G_2.

Zero Inner-product [24]

The ID represents the identity of user which associated with user's private key. Define a vector $\mathbf{X} = (x_1,\ldots,x_n)^{\mathrm{T}}$ such that $x_i = ID^{i-1}$, $i \in [1, n]$. To encrypt with a revoked user set $R = \{ID_1,\cdots, ID_q\}$, one defines as $\mathbf{Y} = (y_1,\ldots, y_n)^{\mathrm{T}}$, the coefficient vector of $P_R[Z]$ from

$$P_R[Z] = \sum_{i=1}^{q+1} y_i Z^{i-1} = \prod_{ID_j \in R} (Z - ID_j)$$

where, if $q + 1 < n$, the coordinates $y_{q+2},\cdots,y_n$ are set to 0. By doing so, we note that $P_R[ID] = <\mathbf{X}, \mathbf{Y}> = 0$ iff $ID \in R$.

For example, if the user ID_1 in the revoked user set $R = \{ID_1, ID_3\}$, we have that $P_R[ID_1] =< \mathbf{X}, \mathbf{Y} >= \prod_{ID_j \in R} (ID_1 - ID_j) = 0$.

Decisional DDH assumption [10]

Let G_1 is a group which prime order is p, let g be a generator of G_1, and give a tuple (g, g^a, g^b) where $a, b \overset{R}{\in} Z_p$, we say that the decisional DDH assumption holds if no polynomial time

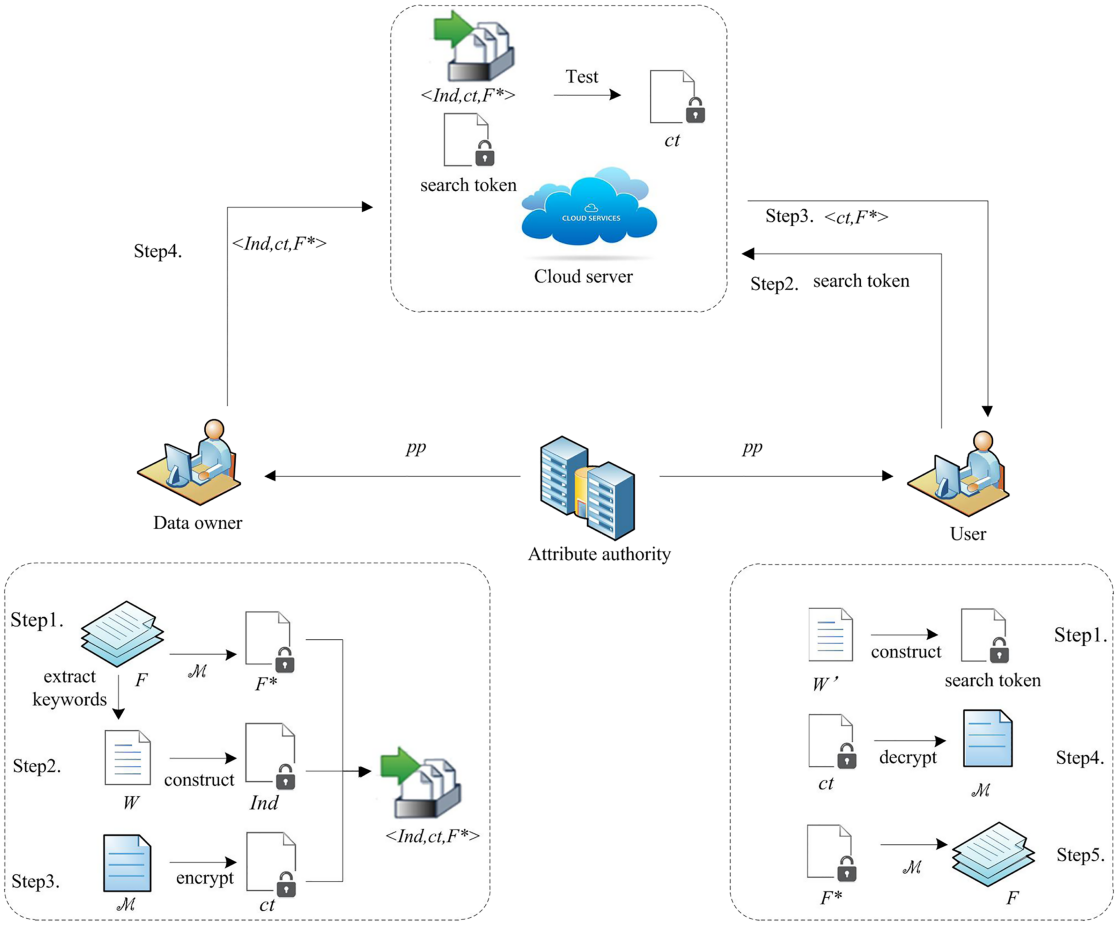

Fig 1. System model of our scheme

https://doi.org/10.1371/journal.pone.0183459.g001

algorithm has a non-negligible advantage to distinguish that Z equals g^{ab} or to a random element of G_1.

Algorithm model and security model

Algorithm model. Denote $U = \{ID_1, \cdots, ID_Q\}$ to be the universe of all the users, we consider a scheme that searchable attribute-based encryption scheme with attribute revocation in cloud storage, as described in Fig 1. There are seven algorithms in our scheme:

Setup $(\lambda) \rightarrow msk, pp$: This algorithm is executed by attribute authority. It inputs a security parameter λ and outputs the master secret key msk and public parameter pp.

KeyGen $(ID, (M, \rho), pp, msk) \rightarrow sk, \tau$:This algorithm is executed by attribute authority. It inputs a user's identity $ID \in U$, an access structure (M, ρ), public parameter pp, the msk and outputs the secret key sk and the part of search token τ.

Encryption $(pp, \omega, R_\theta, m) \rightarrow ct$: This algorithm is executed by data owner. It inputs public parameter pp, the attribute set ω, a revocation list $R_\theta \subseteq U$ which attribute $\theta \in \omega$,a message m and outputs a cipher text ct.

Index $(pp, \omega, R_\theta, W) \rightarrow Ind$: This algorithm is executed by data owner. It inputs public parameter pp, the attribute set ω,a revocation list $R_\theta \subseteq U$ which attribute $\theta \in \omega$,the keywords set from the uploaded files W and outputs keywords index Ind.

Trapdoor $(pp, W', \tau) \rightarrow \tau^*$:This algorithm is executed by user. It inputs the public parameter pp and the keywords set W', and outputs the new token τ^*.

Test $(\tau^*, Ind) \rightarrow 1$ *or* 0:This algorithm is executed by cloud storage server. It inputs the search token $\tau^*$and keywords index Ind and outputs 1 or 0.

Decryption $(pp, ID, sk, R_\theta, ct) \rightarrow m$: This algorithm is executed by user. It inputs public parameter pp, the user secret key sk of user $ID \in U$, a revocation list $R_\theta \subseteq U$ of attribute $\theta \in \omega$, a cipher text ct. And the user ID has the attribute set ω' as: if $ID \in R_\theta$, let $\omega' = \omega - \{\theta\}$;otherwise, $\omega' = \omega$. It computes the message m if and only if the attribute set ω' satisfies the access structure. And the user can decrypt the file with m.

Finally, the system model of our scheme is shown in Fig 1.

Security model

(1) Selective security model of attribute revocation.

Init. The adversary $\mathcal{A}$ chooses the attribute set ω^* and a revocation list $R^*_\theta(\theta \in \omega^*)$.

Setup. The simulator operates this algorithm to get the public parameter pp and sends it to the adversary.

Phase 1. The adversary queries the simulator for user private key sk which corresponds to the access structure (M, ρ), such that $\omega^{*\prime}$will not meet the access structure (M, ρ).

Challenge. The simulator receives two messages m_0 and m_1 from adversary, and chooses a random bit $b \in \{0, 1\}$ to encrypt m_b, and computes challenge cipher text ct^* with the attribute set ω^* and the attribute revocation list R^*_θ.

Phase 2. Same as **Phase 1**.

Guess. The adversary gives a guess b' of b, and the advantage of the adversary in this game is defined as $|\Pr[b' = b] - \frac{1}{2}|$.

Definition1. The game model of this paper is to be safe if there no polynomial time adversaries have a non-negligible advantage in the above game.

(2) Indistinguishability against chosen keyword attack (IND-CKA) model.

Init. The adversary $\mathcal{A}$ selects a attribute set ω^* and a user revocation list R^*_θ of $\theta \in \omega^*$. Then $\mathcal{B}$ runs the algorithm to generate the public parameter pp and sends it to adversary $\mathcal{A}$.

Phase 1. The adversary queries the challenger as follows:

1. The index of keywords $\{w_1, w_2, \ldots, w_N\}$.
2. The search token of $\{w_{j_1}, w_{j_2}, \ldots, w_{j_{N_1}}\}$, and $1 \leq j_1, \ldots, j_{N_1} \leq N$.

Challenge. The challenger receives two different keywords w^*_0 and w^*_1 from the adversary. We require that the keywords w^*_0 and w^*_1 satisfies that $\forall j, w_j \neq w^*_0 \wedge w_j \neq w^*_1$.

The challenger chooses a random keyword w^*_b, $b \in \{0,1\}$, and give the index of keywords w^*_b to adversary.

Phase 2. Same as **Phase 1**.

Guess. The adversary gives a guess b' of b, and the advantage of any adversary in this game is defined as $|\Pr[b' = b] - \frac{1}{2}|$.

Definition 2. We say a searchable encryption article with multiple keywords is secure based on the game IND-CKA, if the advantage of the adversary is negligible in the above game.

Implement of the algorithm

Our construction is based on the Qiang Li et al.[24], and we combine the keyword search with attribute revocation in our new scheme. User constructs the search token when he wants to search files. If the search is successful and the set of attribute satisfies the access structure, it

outputs 1 in the algorithm of Test, then cloud server returns the cipher text. Our scheme adds access control in search, the user can download the files which he interests and can decrypt in this way, and save the space. We construct our scheme as follows:

Setup $(\lambda) \rightarrow msk, pp$: Give that the G_1 and G_2 are two groups of prime order p, the binary size of p is λ,let g be a generator of G_1. Define that e: $G_1 \times G_1 \rightarrow G_2$. In this paper, we suppose the maximum number of attribute is m when encryption, and n represents the maximum number of revoked user set in the revocation list. Then randomly choose $\alpha, \beta, \delta \in Z_p$, $\mathbf{A} = (\alpha_1, \alpha_2, \ldots, \alpha_n)^{\mathrm{T}} \in Z_p^n$, set $\mathbf{H} = (h_1, h_2, \ldots, h_n)^{\mathrm{T}} = (g^{\alpha_1}, g^{\alpha_2}, \ldots, g^{\alpha_n})^{\mathrm{T}}$ and randomly choose $\{k_{0,i}, k_{1,i} \in G_1|_{i=1,\ldots,m}\}$,let $K_0(x) = \prod_{i=1}^{m} k_{0,i}^{(x^i)}$, $K_1(x) = \prod_{i=1}^{m} k_{1,i}^{(x^i)}$. Then randomly choose that $\{t_{0,i}, t_{1,i} \in G_1|_{i=1,\ldots,m}\}$,and then define two functions $T_f(x)$: $Z_p \rightarrow G_1$,$T_f(x) = \prod_{i=1}^{m} t_{f,i}^{(x^i)}$ where $f = \{0, 1\}$. Let hash H be H:$\{0, 1\}^* \rightarrow G_1$, then the master key msk and public parameter pp are:

$$msk = < \alpha, \alpha_1, \beta, \{k_{0,i}, k_{1,i}, t_{0,i}, t_{1,i}\}_{i=1,\ldots,m} >$$

$$pp = < g, e(g,g)^{\alpha}, \mathbf{H} = (h_1, h_2, \ldots, h_n)^{\mathrm{T}}, g^{\beta}, \delta, H, K_0(x), K_1(x) >$$

KeyGen $(ID, (M, \rho), pp, msk) \rightarrow sk, \tau$: Let M be an $l \times k$ matrix corresponding to access policy (M, ρ). Define a vector $\mathbf{X} = (x_1, \ldots, x_n)^{\mathrm{T}}$ such that $x_i = ID^{i-1}$, $i \in [1, n]$. Randomly choose r, $\{z_{i,0}, z_{i,1}\}_{i\in[2,\ldots k]} \in Z_p$, define a vector $\mathbf{v}_0 = (\alpha + r\alpha_1, z_{2,0}, \ldots, z_{k,0})^{\mathrm{T}}$, $\mathbf{v}_1 = (\alpha, z_{2,1}, \ldots, z_{k,1})^{\mathrm{T}}$. For $i = 1$ to l, and compute that $\lambda_{i,0} = M_i \cdot \mathbf{v_0}$ and $\lambda_{i,1} = M_i \cdot \mathbf{v_1}$. Randomly choose $\{r_{i,0}, r_{i,1}\}_{i\in[1,\ldots l]} \in Z_p$, and set the private key as

$$sk = < D_{1,0}, D_{1,1}, D_{2,0}, D_{2,1}, D_3, K_{\mathbf{X}} >$$

where

$$D_{1,0} = \{D_{1,0}^{(i)} = g^{\lambda_{i,0}} T_0(\rho(i))^{r_{i,0}}\}_{i\in[1,\ldots,l]}$$

$$D_{2,0} = \{D_{2,0}^{(i)} = g^{r_{i,0}}\}_{i\in[1,\ldots,l]}$$

$$D_{1,1} = \{D_{1,1}^{(i)} = g^{\lambda_{i,1}} T_1(\rho(i))^{r_{i,1}}\}_{i\in[1,\ldots,l]}$$

$$D_{2,1} = \{D_{2,1}^{(i)} = g^{r_{i,1}}\}_{i\in[1,\ldots,l]}$$

$$D_3 = g^r, K_{\mathbf{X}} = \{K_i = (h_1^{-\frac{x_i}{x_1}} \cdot h_i)^r\}_{i\in[2,\ldots,n]}$$

Then calculate that $K_{\mathbf{X}} = (K_2, \ldots, K_n) = g^{r \cdot M_{\mathbf{X}}^{\mathrm{T}} \mathbf{A}}$, where $M_{\mathbf{X}} \in (Z_p)^{n\times(n-1)}$ is defined by $M_{\mathbf{X}} = \begin{pmatrix} -\frac{x_2}{x_1} & -\frac{x_3}{x_1} & \cdots & -\frac{x_n}{x_1} \\ & I_{n-1} & & \end{pmatrix}$.

Randomly choose $\{v_2, \ldots, v_k\} \in Z_p^{k-1}$ and set $\mathbf{v} = (\beta, v_2, \ldots, v_k)^{\mathrm{T}} \in Z_p^k$. For $i = 1$ to l, compute $\lambda_i = M_i \cdot \mathbf{v}$. Randomly choose $\xi_i \in Z_p$, then denote that

$$\tau = < \tau_1, \tau_{2,0}, \tau_{2,1} >$$

where

$$\begin{aligned}\tau_1 &= \{\tau_{1,i} = g^{\lambda_i}\}_{i=1,\ldots l}\\ \tau_{2,0} &= \{\tau_{2,0}^{\rho(i)} = K_0^{\xi_i}(\rho(i))\}_{i=1,\ldots l}\\ \tau_{2,1} &= \{\tau_{2,1}^{\rho(i)} = K_1^{\xi_i}(\rho(i))\}_{i=1,\ldots l}\end{aligned}$$

then send sk and τ to the user.

Encryption $(pp, \omega, R_\theta, m) \rightarrow ct$: Suppose that a message m is encrypted with a set of attribute ω and a revocation list $R_\theta \subseteq U$ which attribute $\theta \in \omega$. Define a vector $\mathbf{Y} = (y_1, \ldots, y_n)^{\mathrm{T}}$ as the coefficient vector of $P_{R_\theta}[Z]$, and randomly choose $s \in Z_p$ then output

$$ct = \langle C, C_1, C_{2,0}, C_{2,1}, C_3 \rangle$$

where

$$C = m \cdot e(g,g)^{\alpha s}, C_1 = g^s$$

$$C_{2,0} = \{C_{2,0}^{(x)} = T_0(x)^s\}_{x\in\omega}, C_{2,1} = \{C_{2,1}^{(x)} = T_1(x)^s\}_{x\in\omega-\{\theta\}}$$

$$C_3 = (h_1^{y_1} \cdots h_n^{y_n})^s$$

Index $(pp, \omega, R_\theta, W) \rightarrow Ind$: A revocation list $R_\theta \subseteq U$ which attribute $\theta \in \omega$. Data owner encrypts the file F which is firstly encrypted by a symmetric encryption algorithm and gets cipher text F^*, and suppose that the symmetric encryption key is m. The set of keywords $W = \{w_1, w_2, \ldots, w_N\}$ is extracted from the F, and randomly choose $t \in Z_p$,and output the keywords index

$$Ind = < I_0, I_{1,j}, I_{2,0}, I_{2,1} >$$

where

$$I_0 = g^t$$

$$I_{1,j} = g^\beta \cdot H(w_j)^\delta, j \in [1, N]$$

$$I_{2,0} = \{I_{2,0}^{(x)} = K_0^t(x)\}_{x\in\omega},\ I_{2,1} = \{I_{2,1}^{(x)} = K_1^t(x)\}_{x\in\omega-\theta}$$

and send $<Ind, ct, F^*>$ to the cloud server.

Trapdoor $(pp, W', \tau) \rightarrow \tau^*$: The user constructs the search token τ^* according to the keywords $W' = \{w_{j_1}, w_{j_2}, \ldots, w_{j_{N_1}}\}, (1 \le j_1, \ldots, j_{N_1} \le N)$ which he interests as

$$\tau_3 = \{\tau_{1,j_q} = g^\beta \cdot H(w_{j_q})^\delta\}_{q=1,\ldots,N_1 j_q=1,\ldots,N}$$

and sends search token $\tau^* = < \tau_1, \tau_{2,0}, \tau_{2,1}, \tau_3>$ and his ID to the cloud server.

Test $(\tau^*, Ind) \rightarrow 1$ *or* 0: The cloud server receives the search token from the user. First, the cloud server judges that whether the ID of user is in the revocation list R_θ. If $ID \in R_\theta$, let $\omega' = \omega - \{\theta\}$;otherwise, $\omega' = \omega$. If the set ω' satisfies the access structure (M, ρ), then there exists a set of constants $\{\mu_i \in Z_p\}_{i\in I}$, such that $\sum_{i\in I}\mu_i \cdot M_i = (1, 0, \ldots, 0)$.

(1) When $ID \notin R_\theta$, cloud server selects N_1 keywords index from the Ind, we denote the result of selecting as $\{I_{1,O_1}, I_{1,O_2}, \ldots I_{1,O_{N_1}}\}$,where $1 \le O_1, \ldots, O_{N_1} \le N$. Then cloud server tests the selected index set $\{I_{1,O_1}, I_{1,O_2}, \ldots I_{1,O_{N_1}}\}$ with the search token $\tau^* = < \tau_1, \tau_{2,0}, \tau_{2,1}, \tau_3>$ with the

following equation

$$\prod_{q=1}^{N_1} e(I_1, \tau_{1,j_q}) \stackrel{?}{=} \prod_{\sigma=1}^{N_1} e(I_1, I_{1,O_\sigma})$$

If the equation holds, it turns to next step; otherwise, it outputs 0.

$$\frac{e(I_0, \prod_{i\in I}(\tau_{1,i} \cdot \tau_{2,0}^{\rho(i)})^{\mu_i})}{e(\prod_{i\in I}(I_{2,0}^{\rho(i)})^{\mu_i}, g)} \stackrel{?}{=} e(I_0, I_1)$$

If the equations all hold, it returns the corresponding cipher text $<ct, F^*>$ to the user, and user can decrypt. Otherwise, it outputs 0.

(2) When $ID \in R_\theta$, cloud server selects N_1 keywords index from the *Ind*, we denote the result of selecting is $\{I_{1,O_1}, I_{1,O_2}, \ldots I_{1,O_{N_1}}\}$,where $1 \leq O_1, \ldots, O_{N_1} \leq N$. Then cloud server tests the selected index set $\{I_{1,O_1}, I_{1,O_2}, \ldots I_{1,O_{N_1}}\}$ with the search token $\tau^* = < \tau_1, \tau_{2,0}, \tau_{2,1}, \tau_3>$ with the following equation

$$\prod_{q=1}^{N_1} e(I_1, \tau_{1,j_q}) \stackrel{?}{=} \prod_{\sigma=1}^{N_1} e(I_1, I_{1,O_\sigma})$$

If the equation holds, it turns to next step; otherwise, it outputs 0.

$$\frac{e(I_0, \prod_{i\in I}(\tau_{1,i} \cdot \tau_{2,1}^{\rho(i)})^{\mu_i})}{e(\prod_{i\in I}(I_{2,1}^{\rho(i)})^{\mu_i}, g)} \stackrel{?}{=} e(I_0, I_1)$$

If the equations all hold, it returns the corresponding cipher text $<ct, F^*>$ to the user, and user can decrypt. Otherwise, it outputs 0.

Decryption $(pp, ID, sk, R_\theta, ct) \rightarrow m$: User can decrypt according to the returned cipher text. If $ID \in R_\theta$, $\omega' = \omega - \{\theta\}$;otherwise, $\omega' = \omega$, and then:

(1) When $ID \in R_\theta$, let $I = \{i: \rho(i) \in \omega'\}$, and there exists a set of constants $\{\mu_i \in Z_p\}_{i\in I}$, such that $\sum_{i\in I} \mu_i \cdot M_i = (1,0,\ldots,0)$,then $\sum_{i\in I} \mu_i \lambda_{i,1} = \alpha$. It calculates

$$\varphi = \prod_{i\in I}\left(\frac{e(C_1, D_{1,1}^{(i)})}{e(C_{2,1}^{\rho(i)}), D_{2,1}^{(i)})}\right)^{\mu_i} = e(g,g)^{s\alpha}$$

and $m = C / \varphi$, user can decrypt F^* to get F with m.

(2) When $ID \notin R_\theta$, calculate

$$K_{\mathbf{X}} = \prod_{i=2}^{n} K_i^{y_i} = \left(h_1^{-\frac{<\mathbf{X},\mathbf{Y}>}{x_1}} \prod_{i=1}^{n} h_i^{y_i}\right)^r$$

so that when $<\mathbf{X}, \mathbf{Y}> \neq 0$, and then calculate

$$\phi = \left(\frac{e(K, C_1)}{e(C_3, D_3)}\right)^{-\frac{x_1}{<\mathbf{X},\mathbf{Y}>}} = e(g,g)^{rs\alpha_1}$$

Let $I = \{i: \rho(i) \in \omega'\}$, and there exists a set of constants $\{\mu_i \in Z_p\}_{i\in I}$, such that $\sum_{i\in I} \mu_i \cdot M_i = (1,0,\ldots,0)$,then $\sum_{i\in I} \mu \lambda_{i,0} = \alpha + r\alpha_1$. Thus we have

$$\gamma = \prod_{i\in I}\left(\frac{e(C_1, D_{1,0}^{(i)})}{e(C_{2,0}^{\rho(i)}, D_{2,0}^{(i)})}\right)^{\mu_i} = e(g,g)^{s\cdot(\alpha+r\alpha_1)}$$

and $m = C / A$, user can decrypt F^* to get F with m.

Correctness analyses

In this subsection, we show that our construction is correct with some appropriate parameters setting.

(1) In the process of search the equation holds, it means that cloud server selects N_1 keywords index from the *Ind* which we denote $\{I_{1,O_1}, I_{1,O_2}, \ldots I_{1,O_{N_1}}\}$,where $1 \leq O_1, \ldots, O_{N_1} \leq N$ is matching the search token of the keywords $\{w_{j_1}, w_{j_2}, \ldots, w_{j_{N_1}}\}, (1 \leq j_1, \ldots, j_{N_1} \leq N)$ from the user, then computes that

$$\begin{aligned} &\prod_{q=1}^{N_1} e(I_1, \tau_{1,j_q}) \\ &= \prod_{q=1}^{N_1} e(g^{\beta}, g^{\beta} \cdot H(w_{j_q})) \\ &= \prod_{q=1}^{N_1} e(g^{\beta}, I_1 \cdot H(w_{j_q})) \\ &= \prod_{\sigma=1}^{N_1} e(I_1, I_{1,O_\sigma}) \end{aligned}$$

a. When $ID \notin R_\theta$, compute that

$$\begin{aligned} &\frac{e(I_0, \prod_{i \in I}(\tau_{1,i} \cdot \tau_{2,0}^{\rho(i)})^{\mu_i})}{e(\prod_{i \in I}(I_{2,0}^{\rho(i)})^{\mu_i}, g)} \\ &= \frac{e(g^t, g^{\sum_{i \in I} \lambda_i \mu_i} \cdot \prod_{i \in I} K_0^{\xi_i \mu_i}(\rho(i)))}{e(\prod_{i \in I} K_0^{t\xi_i \mu_i}(\rho(i)), g)} \\ &= \frac{e(g^t, g^{\beta}) \cdot e(g, \prod_{i \in I} K_0^{\xi_i \mu_i}(\rho(i)))^t}{e(\prod_{i \in I} K_0^{\xi_i \mu_i}(\rho(i)), g)^t} \\ &= e(g^t, g^{\beta}) \\ &= e(I_0, I_1) \end{aligned}$$

b. When $ID \in R_\theta$, compute that

$$\begin{aligned} &\frac{e(I_0, \prod_{i \in I}(\tau_{1,i} \cdot \tau_{2,1}^{\rho(i)})^{\mu_i})}{e(\prod_{i \in I}(I_{2,1}^{\rho(i)})^{\mu_i}, g)} \\ = &\frac{e(g^t, g^{\sum_{i \in I} \lambda_i \mu_i} \cdot \prod_{i \in I} K_1^{\xi_i \mu_i}(\rho(i)))}{e(\prod_{i \in I} K_1^{t\xi_i \mu_i}(\rho(i)), g)} \\ = &\frac{e(g^t, g^{\beta}) \cdot e(g, \prod_{i \in I} K_1^{\xi_i \mu_i}(\rho(i)))^t}{e(\prod_{i \in I} K_1^{\xi_i \mu_i}(\rho(i)), g)^t} \\ = &e(g^t, g^{\beta}) \\ = &e(I_0, I_1) \end{aligned}$$

(2) The decryption process first calculates

$$K_i = \left(h_1^{-\frac{x_i}{x1}} \cdot h_i\right)^r$$

$$= \left(g^{-\frac{x_i}{x1} \cdot \alpha_1} \cdot g^{\alpha_i}\right)^r$$

$$= g^{r \cdot \left(-\frac{x_i}{x1} \cdot \alpha_1 + \alpha_i\right)}$$

$$M_{\mathbf{X}} = \begin{pmatrix} -\frac{x_2}{x1} - \frac{x_3}{x1} \cdots - \frac{x_n}{x1} \\ I_{n-1} \end{pmatrix}$$

$$\begin{pmatrix} -\frac{x_2}{x1} & \\ -\frac{x_3}{x1} & \\ \vdots & I_{n-1} \\ -\frac{x_n}{x1} & \end{pmatrix} \cdot \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{pmatrix} = \begin{pmatrix} -\frac{x_2}{x1} \cdot \alpha_1 + \alpha_2 \\ -\frac{x_3}{x1} \cdot \alpha_1 + \alpha_3 \\ \vdots \\ -\frac{x_n}{x1} \cdot \alpha_1 + \alpha_n \end{pmatrix} = M_{\mathbf{X}}^{\mathrm{T}} \cdot \mathbf{A}$$

$$K_{\mathbf{X}} = \{K_2, \cdots, K_n\} = g^{r \cdot M_{\mathbf{X}}^{\mathrm{T}} \cdot \mathbf{A}}$$

(3) The decryption process calculates:
a. When $ID \in R_\theta$

$$\begin{aligned}
\varphi &= \prod_{i \in I} \left(\frac{e(C_1, D_{1,1}^{(i)})}{e(C_{2,1}^{\rho(i)}, D_{2,1}^{(i)})}\right)^{\mu_i} \\
&= \prod_{i \in I} \left(\frac{e(g^s, g^{\lambda_{i,1}} T_1(\rho(i))^{r_{i,1}})}{e(T_1(\rho(i))^s, g^{r_{i,1}})}\right)^{\mu_i} \\
&= \prod_{i \in I} \left(\frac{e(g^s, g^{\lambda_{i,1}}) \cdot e(g^s, T_1(\rho(i))^{r_{i,1}}))}{e(T_1(\rho(i))^s, g^{r_{i,1}})}\right)^{\mu_i} \\
&= \prod_{i \in I} (e(g^s, g^{\lambda_{i,1}}))^{\mu_i} \\
&= \prod_{i \in I} e(g, g)^{s \cdot \lambda_{i,1} \cdot \mu_i} \\
&= e(g, g)^{s \cdot (\sum_{i \in I} \lambda_{i,1} \cdot \mu_i)} \\
&= e(g, g)^{s\alpha}
\end{aligned}$$

b. When $ID \notin R_\theta$

$$
\begin{aligned}
K_{\mathbf{X}} &= \prod_{i=2}^{n} K_i^{y_i} \\
&= \prod_{i=2}^{n} \left(h_1^{-\frac{x_i}{x1}} \cdot h_i \right)^{r \cdot y_i} \\
&= \left(h_1^{-\left(\frac{x_2 y_2}{x1} + \cdots + \frac{x_n y_n}{x1}\right)} \cdot \prod_{i=2}^{n} h_i^{y_i} \right)^{r} \\
&= \left(h_1^{-\left(\frac{x_2 y_2}{x1} + \cdots + \frac{x_n y_n}{x1}\right)} \cdot \prod_{i=1}^{n} h_i^{y_i} \cdot h_1^{-y_1} \right)^{r} \\
&= \left(h_1^{-\left(\frac{x_2 y_2}{x1} + \cdots + \frac{x_n y_n}{x1}\right)} \cdot \prod_{i=1}^{n} h_i^{y_i} \cdot h_1^{-\frac{y_1 x_1}{x_1}} \right)^{r} \\
&= \left(h_1^{-\left(\frac{x_1 y_1}{x1} + \cdots + \frac{x_n y_n}{x1}\right)} \cdot \prod_{i=1}^{n} h_i^{y_i} \right)^{r} \\
&= \left(h_1^{-\frac{<\mathbf{X},\mathbf{Y}>}{x1}} \cdot \prod_{i=1}^{n} h_i^{y_i} \right)^{r}
\end{aligned}
$$

$$
\begin{aligned}
\phi &= \left(\frac{e(K, C_1)}{e(C_3, D_3)} \right)^{-\frac{x_1}{<\mathbf{X},\mathbf{Y}>}} \\
&= \left(\frac{e\left(\left(h_1^{-\frac{<\mathbf{X},\mathbf{Y}>}{x1}} \cdot \prod_{i=1}^{n} h_i^{y_i} \right)^{r}, g^s \right)}{e((h_1^{y_1} \cdots h_n^{y_n})^s, g^r)} \right)^{-\frac{x_1}{<\mathbf{X},\mathbf{Y}>}} \\
&= \left(\frac{e\left(\left(h_1^{-\frac{<\mathbf{X},\mathbf{Y}>}{x1}} \right), g \right) \cdot e\left(\left(\prod_{i=1}^{n} h_i^{y_i} \right), g \right)}{e((h_1^{y_1} \cdots h_n^{y_n}), g)} \right)^{-\frac{x_1}{<\mathbf{X},\mathbf{Y}>} r \cdot s} \\
&= \left(e\left(\left(h_1^{-\frac{<\mathbf{X},\mathbf{Y}>}{x1}} \right), g \right) \right)^{-\frac{x_1}{<\mathbf{X},\mathbf{Y}>} r \cdot s} \\
&= \left(e\left(\left(g^{-\frac{<\mathbf{X},\mathbf{Y}>}{x1} \alpha_1} \right), g \right) \right)^{-\frac{x_1}{<\mathbf{X},\mathbf{Y}>} r \cdot s} \\
&= e(g, g)^{rs\alpha_1}
\end{aligned}
$$

$$
\begin{aligned}
\gamma &= \prod_{i\in I}\left(\frac{e(C_1, D_{1,0}^{(i)})}{e(C_{2,0}^{\rho(i)}, D_{2,0}^{(i)})}\right)^{\mu_i} \\
&= \prod_{i\in I}\left(\frac{e(g^s, g^{\lambda_{i,0}} T_0(\rho(i))^{r_{i,0}})}{e(T_0(\rho(i))^s, g^{r_{i,0}})}\right)^{\mu_i} \\
&= \prod_{i\in I}\left(\frac{e(g^s, g^{\lambda_{i,0}}) \cdot e(g^s, T_0(\rho(i))^{r_{i,0}}))}{e(T_0(\rho(i))^s, g^{r_{i,0}})}\right)^{\mu_i} \\
&= \prod_{i\in I}(e(g^s, g^{\lambda_{i,0}}))^{\mu_i} \\
&= \prod_{i\in I} e(g,g)^{s\cdot\lambda_{i,0}\cdot\mu_i} \\
&= e(g,g)^{s\cdot(\sum_{i\in I}\lambda_{i,0}\cdot\mu_i)} \\
&= e(g,g)^{s\cdot(\alpha+r\alpha_1)}
\end{aligned}
$$

Let $A = \gamma / \phi = e(g, g)^{s\alpha}$.

Security analyses

Selective security model proof

Theorem1. If an adversary can break our scheme with advantage ε in the selective security model, then we can construct a simulator to solve the Decision q-BDHE problem with advantage $\frac{\varepsilon}{2}$.

Proof: This proof bases on [24].

The simulation proceeds as follows. First, the challenger sets

$$\mathbf{Y} = (g, g^s, g_1 = g^a, g_2 = g^{a^2}, \ldots, g_q = g^{a^q}, g_{q+2} = g^{a^{q+2}}, \ldots, g_{2q} = g^{a^{2q}})$$

Then the challenger flips a fair binary coin μ: if $\mu = 0$, the challenger sets $Z = e(g_1, g_q)^s$ if $\mu = 1$,then the challenger picks a random element Z from G_2.

Init. The simulator $\mathcal{B}$ runs adversary $\mathcal{A}$. $\mathcal{A}$ selects an attribute set ω^* and a user revocation list R_θ^*,where $\theta \in \omega^*$, which it wishes to be challenged upon.

Setup. The simulator $\mathcal{B}$ proceeds as follows:

(1) The simulator $\mathcal{B}$ randomly chooses $\alpha', \beta, \delta, \in Z_p$, and then simulator $\mathcal{B}$ sets that $e(g,g)^{\alpha} = e(g^a, g^{a^q}) \cdot e(g,g)^{\alpha'}$,implicitly has that $\alpha = \alpha' + \alpha^{q+1}$. Then it randomly chooses $\{k'_{0,i}, k'_{1,i} \in G_1|_{i=1,\ldots,m}\}$, and computes

$$K_0(x) = \prod_{i=1}^{m} k_{0,i}^{\prime(x^i)}, K_1(x) = \prod_{i=1}^{m} k_{1,i}^{\prime(x^i)}$$

(2) It sets $R_\theta^* = \{ID_1, \cdots, ID_m\}$ where $m \le Q$. For $k \in [1, m]$, simulator $\mathcal{B}$ sets $\mathbf{X}_k = (x_{k,1}, \ldots, x_{k,n}) = (1, ID_k, ID_k^2, \ldots, ID_k^{n-1})$, randomly chooses $\mathbf{b}_k \in Z_p$ and has that

$$\mathbf{b}_k^{\mathrm{T}} \cdot M_{\mathbf{X}_k} = \mathbf{b}_k^{\mathrm{T}} \cdot \begin{pmatrix} -\frac{x_{k,2}}{x_{k,1}} \cdots -\frac{x_{k,n}}{x_{k,1}} \\ I_{n-1} \end{pmatrix} = \mathbf{0}$$

and $\mathbf{b}_k = \left(1, \frac{x_{k,2}}{x_{k,1}}, \ldots \frac{x_{k,n}}{x_{k,1}}\right)^{\mathrm{T}}$. The simulator $\mathcal{B}$ sets the $n \times q$ matrix $\mathbf{B} = (\mathbf{b}_1| \ldots |\mathbf{b}_m|\mathbf{0}| \ldots |\mathbf{0})$, for $k \in [1, m]$, it consists by $\mathbf{b}_k$, and $q - m$ columns are $\mathbf{0}$. Sets $\mathbf{Z} = (z_1, \cdots, z_q)^{\mathrm{T}} \in Z^n$ and

$z_i = a^{q+1-i}$, $g^{\mathbf{z}} = (g^{a_q}, \cdots, g^{a})^{\mathrm{T}}$ and implicitly has that $\mathbf{A} = \mathbf{B}\cdot\mathbf{Z} + \boldsymbol{\delta}$ where $\boldsymbol{\delta} \overset{R}{\in} Z_p^n$. Define $\mathbf{H} = (h_1, h_2, \ldots, h_n)^{\mathrm{T}} = g^{\mathbf{B}\cdot\mathbf{Z}}\cdot g^{\boldsymbol{\delta}}$, for $k \in [1, m]$, we have that $M_{\mathbf{X}_k}^{\mathrm{T}} \cdot \mathbf{B} \in (Z_p)^{(n-1)\times q} = \mathbf{0}$, so it doesn't have $z_k = a^{q+1-k}$.

(3) It sets $\omega^{*\prime} = \omega^* - \{\theta\}$, randomly chooses two polynomials $f_0(x)$ and $f_1(x)$ of degree m and computes two polynomials as follows:

$$u_0(x) = x^{m-|\omega^*|}\prod_{i\in\omega^*}(x-i)$$

$$u_1(x) = x^{m-|\omega^*-\{\theta\}|}\prod_{i\in\omega^*-\{\theta\}}(x-i)$$

For $i \in [0, m]$, let $c_{0,i}$ and $c_{1,i}$ be the ith term of $f_0(x)$ and $f_1(x)$, $d_{0,i}$ and $d_{1,i}$ be the ith term of $u_0(x)$ and $u_1(x)$. $\mathcal{B}$ defines $T_0(x) = g^{a\cdot u_0(x)+f_0(x)}$ and $T_1(x) = g^{a\cdot u_1(x)+f_1(x)}$,at the same time, $\mathcal{B}$ simulates $\{t_{0,i}, t_{1,i}\}_{i=1,\ldots,m}$ where

$$t_{0,i} = (g^a)^{d_{0,i}}g^{c_{0,i}}, t_{1,i} = (g^a)^{d_{1,i}}g^{c_{1,i}}$$

Finally, $\mathcal{B}$ gives the public parameters

$$pp = < g, e(g,g)^{\alpha}, \mathbf{H} = (h_1, h_2, \ldots, h_n)^{\mathrm{T}}, g^{\beta}, \delta, K_0(x), K_1(x) >$$

to $\mathcal{A}$.

Phase 1. Let M be a $p\times l$ matrix, $\omega^{*\prime}$ doesn't satisfy the access structure (M, ρ). If $ID \in R_\theta$, there is $\omega^{*\prime} = \omega^* - \{\theta\}$; otherwise, $\omega^{*\prime} = \omega^*$. The simulator $\mathcal{B}$ generates the secret key sk as follows.

(1) When $ID \notin R_\theta$ (in this case, we have $\omega^{*\prime} = \omega^*$), and $\omega^{*\prime}$doesn't satisfy the access structure, $\mathcal{B}$ first defines $\pi = (\pi_1, \cdots, \pi_l)^{\mathrm{T}} \in Z_p^{n*}$ where $\pi_1 = 1$ We have $M_i\cdot\boldsymbol{\pi} = 0$ for each i when $\rho(i) \in \omega^*$. Then the simulator $\mathcal{B}$ defines two vectors $\boldsymbol{\eta}_0 = (r, \eta_{0,2}, \ldots, \eta_{0,l})^{\mathrm{T}}$ and $\boldsymbol{\eta}_1 = (0, \eta_{1,2}, \ldots, \eta_{1,l})^{\mathrm{T}}$, and defines that $\mathbf{u}_0 = \alpha_1\,\boldsymbol{\eta}_0 + \alpha\boldsymbol{\pi}$ and $\mathbf{u}_1 = \boldsymbol{\eta}_1 + \alpha\boldsymbol{\pi}$, we can compute the first term of $\mathbf{u}_0$ and $\mathbf{u}_1$ are $\alpha + r\alpha_1$ and α.

i. When $\rho(i) \in \omega^*$, $\mathcal{B}$ computes that

$$g^{\lambda_{i,0}} = g^{M_i\cdot\mu_0} = (g^{\alpha_1})^{M_i\cdot\eta_0}, g^{\lambda_{i,1}} = g^{M_i\cdot\eta_1}$$

and randomly chooses $r_{i,0}, r_{i,1} \in Z_p$ and computes that

$$D_{1,0}^{(i)} = g^{\lambda_{i,0}}T_0(\rho(i))^{r_{i,0}}, D_{2,0}^{(i)} = g^{r_{i,0}}$$

$$D_{1,1}^{(i)} = g^{\lambda_{i,1}}T_1(\rho(i))^{r_{i,1}}, D_{2,1}^{(i)} = g^{r_{i,1}}$$

ii. When $\rho(i) \notin \omega^*$, $\mathcal{B}$ computes that

$$g^{\lambda_{i,0}} = g^{M_i\cdot\mathfrak{u}_0} = g^{\alpha_1\cdot M_i\cdot\eta_0+\alpha\cdot M_i\cdot\pi}, g^{\lambda_{i,1}} = g^{M_i\cdot\mathfrak{u}_1} = g^{M_i\cdot\eta_1+\alpha\cdot M_i\cdot\pi}$$

and randomly chooses $r, \{r'_{i,0}\}_{i\in[l]}, \{r'_{i,1}\}_{i\in[l]} \in Z_p$, and sets $r_{i,0} = r'_{i,0} - \frac{a^q}{\mu_0(\rho(i))}(M_i\cdot\pi)$ and

$r_{i,1} = r'_{i,1} - \frac{a^q}{\mu_1(\rho(i))}(M_i \cdot \pi)$, then

$$D_{1,0}^{(i)} = g^{\lambda_{i,0}} T_0(\rho(i))^{r_{i,0}} = g^{\alpha_1 \cdot M_i \cdot \eta_0 + \alpha \cdot M_i \cdot \pi} T_0(\rho(i))^{r'_{i,0}} g^{-\frac{a^q \cdot f_0(\rho(i)) \cdot (M_i \cdot \pi)}{u_0(\rho(i))}}$$

$$D_{2,0}^{(i)} = g^{r_{i,0}} = g^{r'_{i,0} - \frac{a^q}{\mu_0(\rho(i))}(M_i \cdot \pi)}$$

$$D_{1,1}^{(i)} = g^{\lambda_{i,1}} T_1(\rho(i))^{r_{i,1}} = g^{M_i \cdot \eta_1 + \alpha \cdot M_i \cdot \pi} T_1(\rho(i))^{r'_{i,1}} g^{-\frac{a^q \cdot f_1(\rho(i)) \cdot (M_i \cdot \pi)}{u_1(\rho(i))}}$$

$$D_{2,1}^{(i)} = g^{r_{i,1}} = g^{r_{i,1} - \frac{a^q}{\mu_1(\rho(i))}(M_i \cdot \pi)}$$

Then $\mathcal{B}$ computes that $D_3 = g^r$, $K_{\mathbf{X}} = \{K_i = (h_1^{-\frac{x_i}{x_1}} \cdot h_i)^r\}_{i \in [2,\ldots,n]}$.

(2) When $ID \in R_\theta^*$ and sets $\{ID = ID_k\}_{k \in [1,m]}$. The simulator $\mathcal{B}$ randomly chooses $r' \in Z_p$ and sets $r = r' - a^k$. Defines $\mathbf{A} = \mathbf{B} \cdot \mathbf{Z} + \boldsymbol{\delta}$, the first term of $\mathbf{A}$ is $\alpha_1 = \delta_1 + \sum_{j=1}^{m} a^{q+1-j}$, and computes that

$$g^{\alpha + r\alpha_1} = g^{\alpha' + a^{q+1}} \cdot \left(g^{\delta_1 + \sum_{j=1}^{m} a^{q+1-j}}\right)^{r' - a^k} = g^{\alpha' - \delta_1 a^k} \cdot g^{\alpha_1 r'} \cdot g^{-\left(\sum_{j=1, j \neq k}^{m} a^{q+1-j+k}\right)}$$

randomly chooses $\{\eta_i\}_{i \in [2,l]} \in Z_p$ and defines $\boldsymbol{\eta} = (\alpha + r\alpha_1, \eta_2, \ldots, \eta_l)^{\mathrm{T}}$, and for $i \in [1, p]$, sets $M_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,l})$, then computes

$$g^{\lambda_{i,0}} = g^{M_i \cdot \boldsymbol{\eta}} = (g^{\alpha + r\alpha_1})^{x_{i,1}} g^{\sum_{j=2}^{l} \eta_j \cdot x_{i,j}}$$

randomly chooses $r_{i,0} \in Z_p$, then

$$D_{1,0}^{(i)} = g^{\lambda_{i,0}} T_0(\rho(i))^{r_{i,0}}, D_{2,0}^{(i)} = g^{r_{i,0}}$$

As $\omega^{*\prime}$ does not satisfy the access structure, the simulation of $D_{1,1}^{(i)}$ and $D_{2,1}^{(i)}$ are the same as the previous case. For $\{K_i\}_{i \in [2,n]}$, the simulator $\mathcal{B}$ can computes $K_{\mathbf{X}} = (K_2, \ldots, K_n) = g^{r \cdot M_{\mathbf{X}}^{\mathrm{T}} \mathbf{A}}$ by $M_{\mathbf{X}}^{\mathrm{T}} \mathbf{A} = M_{\mathbf{X}}^{\mathrm{T}} \cdot \mathbf{B} \cdot \mathbf{Z} + M_{\mathbf{X}}^{\mathrm{T}} \cdot \boldsymbol{\delta}$.

Challenge. The adversary $\mathcal{A}$ submits two messages m_0 and m_1, $\mathcal{B}$ randomly chooses m_b where $b \in \{0,1\}$ to encrypt. Then computes

$$C = m_b \cdot Z \cdot e(g^s, g^{\alpha'}), C_1 = g^s$$

$$C_{2,0} = \{C_{2,0}^{(x)} | C_{2,0}^{(x)} = T_0(x)^s = (g^s)^{f_0(x)}, x \in \omega^*\}$$

$$C_{2,1} = \{C_{2,1}^{(x)} | C_{2,1}^{(x)} = T_1(x)^s = (g^s)^{f_1(x)}, x \in \omega^* - \{\theta\}\}$$

Then the simulator $\mathcal{B}$ defines $\mathbf{Y} = (y_1, \cdots, y_n)^{\mathrm{T}}$ according to the revocation list R_θ^* and $<\mathbf{X}_k, \mathbf{Y}> = 0$ for $k \in [1,m]$. And we have that $\mathbf{Y} = M_{X_k} \cdot \boldsymbol{\gamma}_1$ where $\boldsymbol{\gamma}_1 = (y_2, \cdots, y_n)^{\mathrm{T}}$, then

$$< \mathbf{Y}, \mathbf{B} \cdot \mathbf{Z} >= \mathbf{Y}^{\mathrm{T}}\mathbf{B} \cdot \mathbf{Z} = \sum_{\mathrm{k}=1}^{m} z_k \cdot \mathbf{Y}^{\mathrm{T}} \cdot \mathbf{b}_k = 0$$

and computes

$$C_3 = (h_1^{y_1} \ldots h_n^{y_n})^s = (g^s)^{<\mathbf{Y},\mathbf{A}>} = (g^s)^{<\mathbf{Y},\boldsymbol{\delta}>}$$

Then $\mathcal{B}$ sends the challenge ciphertext $ct^* = (C, C_1, C_{2,0}, C_{2,1}, C_3)$ to the adversary $\mathcal{A}$. If $\mu = 0$, then $Z = \mathrm{e}(g_1, g_q)^s$, the challenge ciphertext ct^* is a valid random encryption of message m_b. If $\mu = 1$, then Z is a random element of G_2, and ct^* is also random from the adversary's view, and ct^* contains no information of m_b.

Phase2. Same as **Phase1**.

Guess. The adversary $\mathcal{A}$ outputs the guess b' of b. $\mathcal{B}$ outputs $\mu = 0$ to guess that $Z = e(g_1, g_q)^s$ if $b' = b$; otherwise, $\mathcal{B}$ outputs $\mu = 1$, and it indicates that Z is a random element in G_2. And the advantage of simulator $\mathcal{B}$ to solve the q-BDHE problem is

$$\begin{aligned} & \frac{1}{2}\Pr[\mu' = \mu | \mu = 0] + \frac{1}{2}\Pr[\mu' = \mu | \mu = 1] - \frac{1}{2} \\ = \; & \frac{1}{2}(\frac{1}{2} + \varepsilon) + \frac{1}{2} \cdot \frac{1}{2} - \frac{1}{2} \\ = \; & \frac{\varepsilon}{2} \end{aligned}$$

IND-CKA security proof

Theorem 2. Suppose there exists a polynomial-time adversary $\mathcal{A}$, which can attack our scheme with advantage ε in the IND-CKA model. We can construct a simulator $\mathcal{B}$ that can solve the DDH problem in G_1 with probability at lest $\frac{\varepsilon}{4e(M+TN_1+\frac{1}{2})}$, where e is constant, and we assume the adversary $\mathcal{A}$ makes M index queries and T search token queries(it contains N_1 keywords) in each phase[10].

Proof: $\mathcal{B}$ is given an instance g, g^a, g^b, g^c of the DDH problem in G_1. In the following parts, we construct the cipher text by setting $\delta = b$. The simulation proceeds as follows:

Init. The adversary $\mathcal{A}$ selects a attribute set ω^* and a user revocation list R_θ^* of $\theta \in \omega^*$.$\mathcal{B}$ is given an instance g, g^a, g^b, g^c of the DDH problem in G_1. Then $\mathcal{B}$ runs the algorithm to generate the public parameter pp and sends it to adversary $\mathcal{A}$.

Phase1. $\mathcal{B}$ maintains a hash list $L = \{w_j, \alpha_j, l_j\}$ and randomly chooses $\alpha_j \in Z_p$ for keywords w_j with biased coin flip l_j. The list is empty when begins and simulates the hash function as a

random oracle. And if the random oracle is queried for a hash of w,$\mathcal{B}$ searches the hush list L if the w exists in the list.

1. If $l_j = 0$,the $\mathcal{B}$ gives that g^{α_j};
2. If $l_j = 1$,the algorithm aborts;
3. If the keyword w does not exist in the list, the $\mathcal{B}$ flips a random coin $l \in \{0,1\}$ so that Pr $[\text{coin}' = 0] = \sigma$ and σ will be calculated later.
   a. If $l = 0$, the $\mathcal{B}$ randomly chooses $\alpha \in Z_p$,and adds $< w, \alpha, 0 >$ to the hush list;
   b. If $l = 1$, the $\mathcal{B}$ adds $< w, \perp, 1 >$ to the hush list.
   c. The $\mathcal{B}$ repeat the above process.

Keywords index query. If the adversary $\mathcal{A}$ asks the keyword w_j of index information, $\mathcal{B}$ searches the hush list L. If $l_j = 1$, $\mathcal{B}$ aborts; and if $l_j = 0$, $\mathcal{B}$ randomly chooses $t \in Z_p$, let $H(w_j) = g^{\alpha_j}$ and generates that

$$I_0 = g^t$$

$$I_{1,j} = g^{\beta} H(w_j)^{\delta} = g^{\beta}(g^b)^{\alpha_j}$$

$$I_{2,0} = \{I_{2,0}^{(x)} = K_0^t(x)\}_{x \in \omega^*}, I_{2,1} = \{I_{2,1}^{(x)} = K_1^t(x)\}_{x \in \omega^* - \theta}$$

Search token query. If the adversary $\mathcal{A}$ asks the keyword w_{j_q} of searching token with the access structure (M, ρ), Let M be a $p \times l$ matrix, $\omega^{*\prime}$doesn't satisfy the access structure (M, ρ). If $ID \in R_\theta^*$, there is $\omega^{*\prime} = \omega^* - \{\theta\}$; otherwise, $\omega^{*\prime} = \omega^*$.$\mathcal{B}$ searches the hush list L. If $l_{j_q} = 1$,$\mathcal{B}$ aborts; and if $l_{j_q} = 0$,let $H(w_{j_q}) = g^{\alpha_j}$. For $i = 1$ to l, randomly choose $\xi_i \in Z_p$ and $\mathcal{B}$ generates that

$$\begin{aligned}
\tau_1^* &= \{\tau_{1,i,j_q} = g^{\lambda_i} H(w_{j_q})^{\delta}\}_{i \in [1,l], q \in [1,N_1], j_q \in [1,N]} \\
\tau_{2,0} &= \{\tau_{2,0}^{\rho(i)} = K_0^{\xi_i}(\rho(i))\}_{i \in [1,l]} \\
\tau_{2,1} &= \{\tau_{2,1}^{\rho(i)} = K_1^{\xi_i}(\rho(i))\}_{i \in [1,l]}
\end{aligned}$$

Challenge. The adversary $\mathcal{A}$ outputs two keywords w_0^* and w_1^*,$\mathcal{B}$ randomly chooses $b \in \{0,1\}$ and searches the hush list L that $< w_b^*, \alpha, l >$. If $l = 0$,$\mathcal{B}$ aborts; if $l = 1$, let $H(w_b^*) = g^a$ and computes

$$I_0 = g^t, I_1 = g^{\beta} g^c$$

$$I_{2,0} = \{I_{2,0}^{(x)} = K_0^t(x)\}_{x \in \omega^*}, I_{2,1} = \{I_{2,1}^{(x)} = K_1^t(x)\}_{x \in \omega^* - \theta}$$

Phase2. Same as **Phase1**.

Guess. The adversary $\mathcal{A}$ outputs the guess b' of b, $\mathcal{B}$ outputs $g^c = g^{ab}$ if $b' = b$; otherwise g^c is a random group element in G_1.

Correctness Analyses. In the above simulation scheme, if the adversary $\mathcal{A}$ has the advantage of attack our scheme, and then it will be given the keyword w_j of hush value is $H(w_j) = g^a$ rather than the random value $H(w_j) = g^{aj}$. Then it can compute that $I_1 = g^{\beta} H(w)^{\delta} = g^{\beta}(g^b)^a$, that is $I_1 = g^{\beta} g^c = g^{\beta} g^{ab}$, and $\mathcal{B}$ computes that $g^c = g^{ab}$ which means it solves the DDH problem.

Table 1. Performance analyses.

Scheme	Fine-grained	Attribute revocation	Keyword search	Do not update cipher-text when attribute revocation
[26]	×	×	×	×
[21]	×	×	√	×
[24]	√	√	×	√
[27]	×	×	×	_
[28]	×	×	×	_
Our scheme	√	√	√	√

https://doi.org/10.1371/journal.pone.0183459.t001

Probability Analyses. Suppose that the adversary $\mathcal{A}$ makes M index queries and T search token queries in each phase, and the probability that $\mathcal{B}$ will not be terminated in two query phases 1 and 2 is $\sigma^{2(M+TN_1)}$, so the probability that it will not terminated during the challenge step is $1-\sigma$, so that results in an overall probability that $\mathcal{B}$ does not abort is $\sigma^{2(M+TN_1)} \cdot (1-\sigma)$. And, through the computes that the maximum is $\sigma = 1 - \frac{1}{2(M+TN_1)+1}$, so the maximum probability is $\frac{1}{2e(M+TN_1+\frac{1}{2})}$. Thus, if our scheme can be attacked by the adversary $\mathcal{A}$ with the advantage ε, and the $\mathcal{B}$ can resolve the DDH problem with advantage $\frac{\varepsilon}{4e(M+TN_1+\frac{1}{2})}$.

Performance analyses

In this section, we give some performance analysis in our scheme. The hardware runtime environment is Intel Core i5-3470 CPU @ 3.20GHz, and RAM is 4.00GB. The software runtime environment is JDK 1.7.5, JPBC 2.0.0 and MyEclipse10.

Our scheme is compared with the schemes of [21, 24, 26, 27, 28] in Table 1.

Our scheme is also compared with the schemes of [26, 27, 28] in Table 2.

We can see from Table 2, our scheme has a large amount of computation in the KenGen and Encryption generation, because our scheme doesn't need to update the cipher-text and secret key when attributes revocation. However, the schemes of [26], [27] and [28] don't achieve the function of attribute revocation.

As is shown in the Fig 2, we suppose that there are 16 attributes in the policy and provide the relational graphs of keywords index building time as is shown in Fig 2(a) and search token building time as is shown in Fig 2(b). From the Fig 2(a) and 2(b), we can see that the time cost is nearly linear with the index building and token building. In the Fig 2(c), we give the relational graph of the number of attributes in the policy and time cost. As is shown in the Fig 2(c),

Table 2. Calculation analyses.

Scheme	KeyGen	Encryption	Pairings in Decryption
[26]	$(2+2l)ex$	$(3+\lvert S\rvert)ex$	$2+2\lvert I\rvert$
[27]	$3lex$	$(2+\lvert S\rvert)ex$	$1+3\lvert I\rvert$
[28]	$2lex$	$(6+\lvert S\rvert)ex$	$1+2\lvert I\rvert$
Our scheme	$(2+4l)ex$	$(3+2\lvert S\rvert)ex$	$1+2\lvert I\rvert$

$|S|$: The size of the attributes set of a decryption key.

l: The number of rows of the matrix in access policy(M,ρ).

ex: An exponentiation operation.

$|I|$: The number of attributes for a decryption key to satisfy a cipher-text policy.

https://doi.org/10.1371/journal.pone.0183459.t002

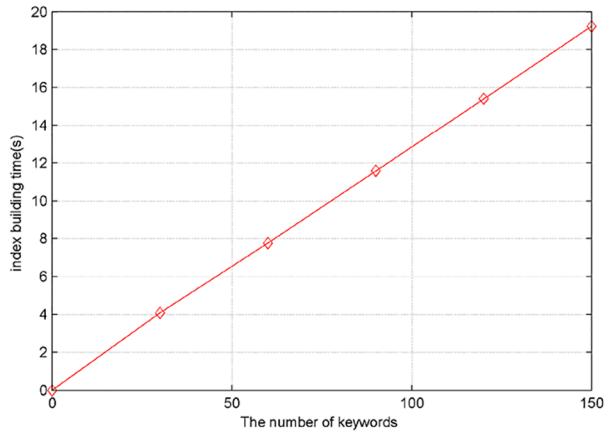

Fig2.(a) Index building time

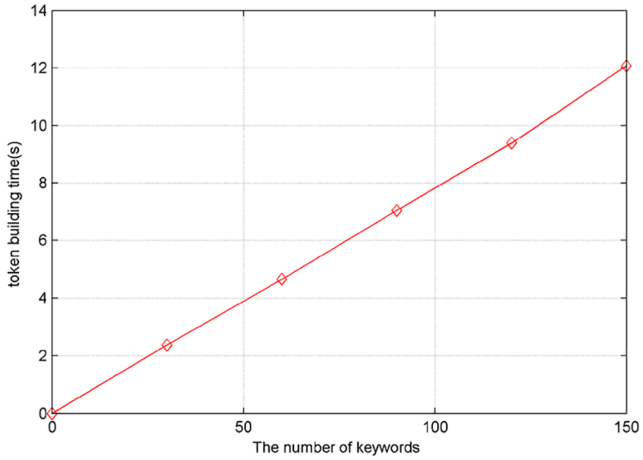

Fig2.(b) Token building

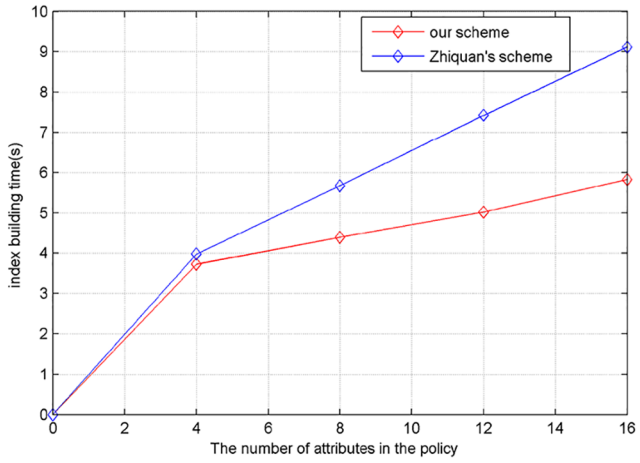

Fig2.(c) The number of attributes in policy and index building

Fig 2. (a) Index building time (b) Token building time (c) The number of attributes in policy and index building time

https://doi.org/10.1371/journal.pone.0183459.g002

we can find that the effect of the increase of the attributes on the time is not particularly evident in our scheme which takes less time than Zhiquan's[29].

Conclusions

In our scheme, we add the keyword search based on the attribute revocation, the search tokens generated by the attribute authority and the user. The cloud server match is divided into two cases: the user is in the revocation list and not in the revocation list, and the cloud server uses the different test according to the different case. It will return the cipher text when the attribute set meets the access structure and the search keywords exist, and the user can decrypt correctly. This scheme supports multiple keywords search at the same time which makes more flexible in the practical application.

Supporting information

S1 Appendix.
(RAR)

Acknowledgments

This work is supported by the National Natural Science Foundation of China under grants 61572019, 61173192, the Key Project of Research Foundation of Natural Science Foundation of Shaanxi Province of China under Grant No. 2016JZ001. Thanks also go to the anonymous reviewers for their useful comments.

Author Contributions

Writing – original draft: Shangping Wang, Duqiao Zhao.

Writing – review & editing: Yaling Zhang.

References

1. Sahai Amit, and Waters B.. Fuzzy Identity-Based Encryption. Advances in Cryptology–EUROCRYPT 2005. Springer Berlin Heidelberg, 2005:457–473.
2. Pirretti M, Traynor P, Mcdaniel P, et al. Secure attribute-based systems. IOS Press, 2006:99–112.
3. Boldyreva A, Goyal V, Kumar V. Identity-based encryption with efficient revocation. ACM Conference on Computer and Communications Security. ACM, 2008:417–426.
4. Hinek MJ, Jiang S, Safavi-Naini R, Shahandashti SF. Attribute-based encryption with key cloning protection. Bulletin of the Korean Mathematical Society. 2008; 2008(4):803–19.
5. Li J, Ren K, Kim K. A2BE: Accountable Attribute-Based Encryption for Abuse Free Access Control. Iacr Cryptology Eprint Archive. 2009; 2009.
6. Attrapadung N, Imai H. Conjunctive Broadcast and Attribute-Based Encryption. Pairing-Based Cryptography—Pairing 2009, Third International Conference, Palo Alto, CA, USA, August 12–14, 2009, Proceedings. DBLP, 2009:248–265.
7. Touati L, Challal Y. Batch-based CP-ABE with attribute revocation mechanism for the Internet of Things. International Conference on Computing, NETWORKING and Communications. IEEE, 2015:1044–1049.
8. Wang PP, Feng DG, Zhang LW. CP-ABE Scheme Supporting Fully Fine-Grained Attribute Revocation. Journal of Software. 2012; 23(10):2805–2816.
9. Boneh D, Crescenzo G D, Ostrovsky R, et al. Public Key Encryption with Keyword Search. Advances in Cryptology—EUROCRYPT 2004. Springer Berlin Heidelberg, 2004:506–522.
10. Kerschbaum F. Secure conjunctive keyword searches for unstructured text. International Conference on Network and System Security, Nss 2011, Milan, Italy, September. DBLP, 2011:285–289.

11. Cao N, Wang C, Li M, Ren K, Lou W. Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data. IEEE Transactions on Parallel & Distributed Systems. 2014; 25(1):222–233. https://doi.org/10.1016/j.jbiomech.2005.09.015
12. Chuah M, Hu W. Privacy-Aware BedTree Based Solution for Fuzzy Multi-keyword Search over Encrypted Data. International Conference on Distributed Computing Systems Workshops. IEEE Computer Society, 2011:273–281.
13. Han F, Qin J, Zhao H, Hu J. A general transformation from KP-ABE to searchable encryption. Future Generation Computer Systems. 2014; 30(1):107–115.
14. Chung KM, Kalai Y, Vadhan S. Improved Delegation of Computation Using Fully Homomorphic Encryption: Springer Berlin Heidelberg; 2010. 483–501 p.
15. Gentry C. Fully Homomorphic Encryption Using Ideal Lattices. Proceedings of the Annual Acm Symposium on Theory of Computing. 2009; 9(4):169–78.
16. Liang K, Susilo W. Searchable Attribute-Based Mechanism with Efficient Data Sharing for Secure Cloud Storage. IEEE Transactions on Information Forensics and Security. 2015; 10(9):1981–92. https://doi.org/10.1109/TIFS.2015.2442215
17. Li H, Yang Y, Luan TH, Liang X, Zhou L, Shen XS. Enabling Fine-Grained Multi-Keyword Search Supporting Classified Sub-Dictionaries over Encrypted Cloud Data. IEEE Transactions on Dependable and Secure Computing. 2016; 13(3):312–25. https://doi.org/10.1109/TDSC.2015.2406704
18. Liang K, Susilo W. Searchable Attribute-Based Mechanism with Efficient Data Sharing for Secure Cloud Storage. IEEE Transactions on Information Forensics & Security. 2015; 10 (9):1981–1992.
19. Li J, Shi Y, Zhang Y. Searchable ciphertext-policy attribute-based encryption with revocation in cloud storage. International Journal of Communication Systems. 2017, 30 (1).
20. Sun W, Yu S, Lou W, Hou YT, Li H. Protecting Your Right: Verifiable Attribute-Based Keyword Search with Fine-Grained Owner-Enforced Search Authorization in the Cloud. IEEE Transactions on Parallel and Distributed Systems. 2016; 27(4):1187–98. https://doi.org/10.1109/TPDS.2014.2355202
21. Yang Y, Ma M. Conjunctive Keyword Search with Designated Tester and Timing Enabled Proxy Re-Encryption Function for E-Health Clouds. IEEE Transactions on Information Forensics and Security. 2016; 11 (4):746–759. https://doi.org/10.1109/TIFS.2015.2509912
22. Jiang X, Yu J, Yan J, Hao R. Enabling efficient and verifiable multi-keyword ranked search over encrypted cloud data. Information Sciences. 2017; s 403–404:22–41.
23. Poon HT, Miri A, editors. A Combined Solution for Conjunctive Keyword Search, Phrase Search and Auditing for Encrypted Cloud Storage. Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress; 2017.
24. Li Q, Feng D, Zhang L. An attribute based encryption scheme with fine-grained attribute revocation. Global Communications Conference (GLOBECOM), 2012 IEEE. 2012:885–890.
25. Shi Y, Zheng Q, Liu J, Han Z. Directly revocable key-policy attribute-based encryption with verifiable ciphertext delegation. Information Sciences. 2015; 295:221–231.
26. Zhang M, Du W, Yang X, Han Y. A fully secure KP-ABE scheme in the standard model. Journal of Computer Research & Development. 2015.
27. Li Z, Chen X. Attribute-based encryption with fast decryption on prime order groups. Computer application. 2016; 36 (3):637–641.
28. Ma S, Lai J, Deng RH, Ding X. Adaptable key-policy attribute-based encryption with time interval. Soft Computing. 2016:1–10.
29. Lv Z, Zhang M, Feng D. Multi-user Searchable Encryption with Efficient Access Control for Cloud Storage. IEEE International Conference on Cloud Computing Technology and Science. IEEE, 2015:366–373.