

# A key-insulated CP-ABE with key exposure accountability for secure data sharing in the cloud

**Hanshu Hong<sup>1</sup>, Zhixin Sun<sup>1</sup>, Ximeng Liu<sup>2</sup>**

<sup>1</sup> Key Lab of Broadband Wireless Communication and Sensor Network Technology, Nanjing University of Posts and Telecommunications  
Nanjing, China

[e-mail: 2014070244@njupt.edu.cn]

<sup>1</sup> Key Lab of Broadband Wireless Communication and Sensor Network Technology, Nanjing University of Posts and Telecommunications

[e-mail: sunzx@njupt.edu.cn]

<sup>2</sup> School of information systems, Singapore Management University

[e-mail: xmliu@smu.edu.sg]

\*Corresponding author: Zhixin Sun

*Received January 1, 2016; revised March 17, 2016; accepted April 6, 2016;  
published May 31, 2016*

---

## **Abstract**

ABE has become an effective tool for data protection in cloud computing. However, since users possessing the same attributes share the same private keys, there exist some malicious users exposing their private keys deliberately for illegal data sharing without being detected, which will threaten the security of the cloud system. Such issues remain in many current ABE schemes since the private keys are rarely associated with any user specific identifiers. In order to achieve user accountability as well as provide key exposure protection, in this paper, we propose a key-insulated ciphertext policy attribute based encryption with key exposure accountability (KI-CPABE-KEA). In our scheme, data receiver can decrypt the ciphertext if the attributes he owns match with the self-centric policy which is set by the data owner. Besides, a unique identifier is embedded into each user's private key. If a malicious user exposes his private key for illegal data sharing, his identity can be exactly pinpointed by system manager. The key-insulation mechanism guarantees forward and backward security when key exposure happens as well as provides efficient key updating for users in the cloud system. The higher efficiency with proved security make our KI-CPABE-KEA more appropriate for secure data sharing in cloud computing.

---

**Keywords:** Key-insulated, ABE, accountability, key exposure protection, secure

## 1. Introduction

With the technical development of distributed storage and virtualization, more and more users tend to move their data to the cloud [1]. The cloud center contains massive private data of users, hence special encryption techniques should be implemented to protect the privacy and confidentiality of these sensitive data from being attacked. However, traditional cryptosystem can hardly meet a series of new demands which emerge in cloud computing systems. Under this situation, attribute based encryption (ABE) [2] is proposed and has been divided into two mechanisms: KP-ABE [6-7] [23-24] and CP-ABE [3] [22]. Among the two kinds of ABE mechanisms, CP-ABE is considered more appropriate in terms of data protection in cloud computing, since it is equipped with the ability of providing flexible self-centric data access management. In CP-ABE, the private keys of a user corresponds with the attributes he owns, while the ciphertext corresponds a self-centric access policy which is made by the data owner. A data receiver is capable of decrypting the encrypted file provided that the attributes he possesses match with the policy.

However, key abuse [19] is a challenging issue in attribute based cryptosystem. ABE is an advanced type of broadcast encryption, users possessing the same attributes share the same private keys. There may exist some traitors expose their private keys illegally without being detected, which will threaten the security of the whole system. Key abuse issues remain in many current ABE schemes as the private keys generated are rarely associated with any users' specific identifiers. One method is to assign each user a unique pair of attribute public key and private key, when private key exposure happens, the identity of traitor can easily be traced. However, if this mechanism is introduced, the encryption cost will be enormous and the advantages of ABE will no longer exist. Thus, it is essential to propose a scheme which takes the advantage of ABE as well as meets the demands of distinguishing different users by the private keys they possess.

Existing researches concerning ABE have achieved much progress with regard to fine-grained access control [4-5] [15], flexible attribute revocation [9] [14], data authentication [8] [16], etc. However, key exposure accountability and key exposure protection in ABE have not received much concern. Some schemes have successfully achieved user accountability in ABE [10-13], but the computation cost and the size of private keys in their schemes are too large, which make them impractical to be totally applied to some sceneries with constrained computing resources such as wireless networks, mobile cloud computing, etc. Besides, although schemes in [10-13] are capable of tracing the malicious user when key exposure happens, the system will still be at risk since there doesn't exist effective key updating mechanisms to guarantee forward and backward security. If the private key owned by valid user is leaked, any other user obtaining it can decrypt the corresponding ciphertext since the leaked private key is still a valid one. Consequently, all the potential threat calls for frequent key refreshing in attribute based cryptosystem. When key exposure happens, effective and efficient key updating mechanism should be implemented to keep the system from potential threat. Consequently, a scheme equipped with both user accountability and secure key updating is urgently to be put forward.

To better tackle the issues discussed above, in this paper, we propose a key-insulated ciphertext based attribute based encryption with key exposure accountability (KI-CPABE-KEA) scheme. In our scheme, a unique identifier is embedded into each user's private key. If a malicious user leaks his private key for illegal data sharing, the system manager can pinpoint the exact identity of the traitor. We introduce key-insulation mechanism into our scheme and divide the system lifetime into several discrete time periods. When time

period evolves, only part of the private keys have to be updated and the system public parameters remain unchanged. Due to the key-insulation mechanism, our scheme is very efficient with regard to key updating computation cost as well as provides key exposure protection when key leakage happens. Moreover, our KI-CPABE-KEA is proved to be secure under DBDH assumption. The higher performance makes our scheme more appropriate for secure data protection in cloud computing.

The rest of paper is arranged as follows:

Section 2 reviews relevant research related to ABE with user accountability and gives the background of some fundamental preliminaries which are used to construct our scheme. Section 3 illustrates the model of our KI-CPABE-KEA and makes some essential security definitions. Section 4 contains a full description of the algorithms in KI-CPABE-KEA along with the correctness proof. Section 5 presents the security analysis and evaluates our scheme with respect to computation efficiency. In section 6, the conclusion of this paper and future prospects are made.

## 2. Related works and preliminaries

### 2.1 Related works

The notion of ABE was first put forward in [1]. In ABE, user's access privileges are described by a number of attributes rather than an identity string. A user can get access to the ciphertext only if his attributes match with the access structure which is made by the data owner. Equipped with the advantages of providing flexible data control, ABE is widely implemented as an effective tool for secure data sharing between users in different application scenarios [14] [18] [20].

Key exposure is a challenging problem in ABE. M. J. Hinek et al. in [10] first proposed a token-based ABE which provides key exposure accountability. In their scheme, user's keys consists of two parts: private keys and delegation keys. User's delegation key is embedded with user's unique identifier, but the contribution of delegation agent will add more computation cost to the whole system. Jin et al. in [11] tackled the key exposure problem by proposing an accountable and anonymous CP-ABE scheme. In their scheme, the identity of the traitor can be traced in a black-box model. Their scheme is provably secure under DBDH assumption. Based on [11], Fatos et al. in [12] achieved user accountability in CP-ABE and apply their scheme to PHR systems. In their scheme, a unique identifier is inserted into user's decryption key, if a malicious PHR user leaks his private key deliberately to other unauthorized users, the identity of the malicious user can be exactly traced. Wang et al. in [13] proposed a key-policy ABE with user accountability. The identity of traitor can be traced by a white-box setting, but the overhead of encryption and decryption is too large.

Besides, although the above schemes have achieved user accountability, they haven't taken backward and forward security into full consideration when key exposure happens, which may bring serious security risks to the system. If the private key owned by a malicious user is leaked, any user obtaining the private key can decrypt the corresponding ciphertext since the leaked private key is a valid one. Thus, when key exposure happens, the identity of the malicious user should be exactly pinpointed by the system manager; more importantly, effective and efficient key updating mechanism should be implemented to keep the system secure. The mechanism of key-insulation was initially proposed in [21]. Key-insulation is a promising tool to guarantee forward and backward security as well as achieves high efficiency of key updating. In this mechanism, the lifespan of a cryptosystem is partitioned by several discrete time periods. The public parameters remain unaltered during the whole time periods, while private keys are updated when system enters into a new time period.

Key-insulation mechanism can provide full security when user’s private key exposure happens, which can also be introduced to provide key exposure protection in attribute based cryptosystem[25].

### 2.2 Hardness assumptions

a. Discrete Logarithm Assumption (DL):

Given  $X, P \in G$ , it is computational infeasible to calculate the value of  $a$  ( $a \in Z_q^*$ ) such that  $X = P^a$  with a non-negligible probability within probabilistic polynomial-time.

b. Decision Bilinear Diffie-Hellman problem (DBDH).

For  $a, b, c, z \in Z_q^*$ , given  $(g, g^a, g^b, g^c, z)$ , it is computational infeasible to distinguish the following tuples  $(A = g^a, B = g^b, C = g^c, \hat{e}(g, g)^{abc})$  and  $(A = g^a, B = g^b, C = g^c, \hat{e}(g, g)^z)$  with non-negligible probability within probabilistic polynomial-time.

## 3. Models and assumptions

### 3.1 The model of our KI-CPABE-KEA

As is shown in Fig. 1, the model includes 6 entities: CSP (cloud service provider), AA (attribute authority), key helper, tracer, data owner and data receiver.

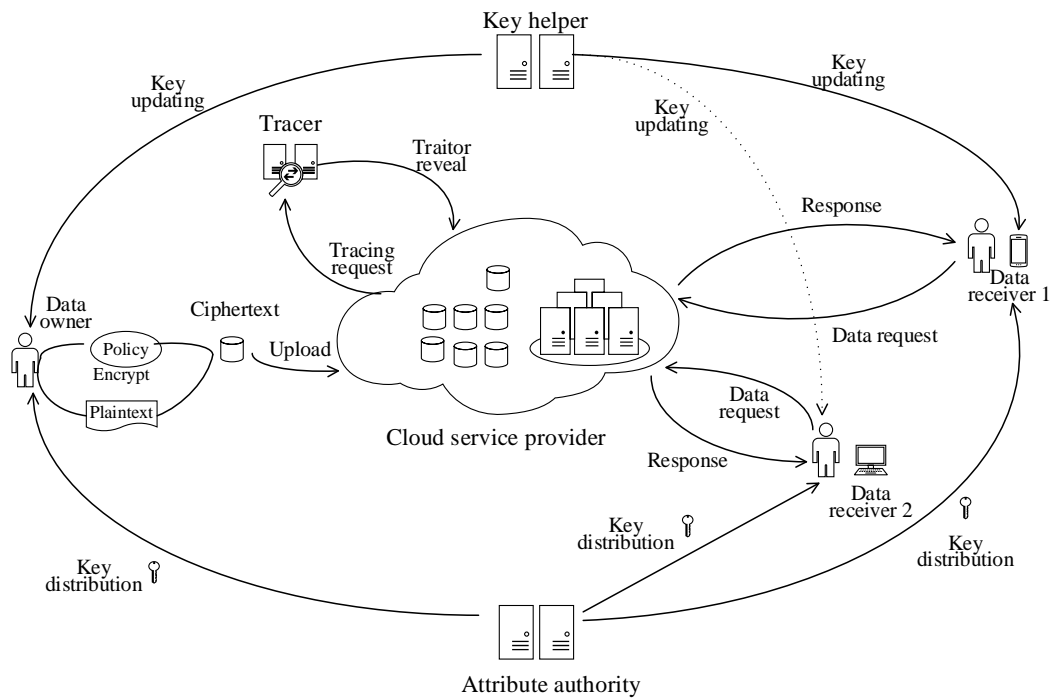


Fig. 1. The model of our KI-CPABE-KEA

CSP assigns a global unique identifier to each user in the system and responses to the data requests made by users. AA manages the global attributes in the system. Meanwhile, AA

generates initial private key for a user according to the attribute set and the unique identifier he owns. Key helper generates the key updating component for each user when system evolves into a new time period. Data owner encrypts his file using a policy and transfers the encrypted data to CSP. Data receiver sends access request to CSP and decrypts the ciphertext under the condition that the attributes he owns match with the encryption policy. When a malicious user in the system leaks his private key deliberately for the purpose of illegal data sharing, tracer can pinpoint the exact identity of the malicious user according to the exposed private key.

### 3.2 Formulated definitions of algorithms

Our KI-CPABE-KEA includes five algorithms.

$Setup(1^\varphi) \rightarrow \{PK, MK, HK\}$  : This algorithm takes a parameter  $\varphi$  as input, it returns the system public parameter  $PK$ , master key  $MK$  and helper key  $HK$ .

$Initial\ private\ key\ generation\{PK, MK, \{A_i\}, id, TP_0\} \rightarrow \{SK_{id, TP_0}\}$  : On input the system public key  $PK$ , the master key  $MK$ , initial time period  $TP_0$  and a user's  $id$  with attribute set  $\{A_i\}$ , AA outputs  $SK_{id, TP_0}$  as the initial private key of a user.

$Key\ update\{TP_n, TP_{n+1}, HK, \{A_i\}, SK_{id, TP_n}\} \rightarrow \{UP_{i, TP_{n+1}}, SK_{id, TP_n}\}$ : The algorithm takes two time periods  $TP_n, TP_{n+1}$ , attribute set  $\{A_i\}$  and  $HK$  as input, it outputs the key updating component  $UP_{i, TP_{n+1}}$ . Users update their temporal private keys to the lasted version by calculating  $SK_{TP_{n+1}} = SK_{id, TP_n} \cdot UP_{i, TP_{n+1}}$ .

$Encrypt\{PK, M, \gamma\} \rightarrow \{CT\}$  : This algorithm takes the systems public parameter  $PK$ , a plaintext  $M$ , and an encryption policy  $\gamma$  as input. Then the algorithm outputs a ciphertext  $CT$ .

$Decrypt\{SK_{id, TP_n}, CT\} \rightarrow \{M\}$ : The algorithm takes  $CT$  and receiver's temporal private key  $SK_{id, TP_n}$  as input, it outputs the plaintext  $M$ .

### 3.3 Security model

Definition: Our KI-CPABE-KEA has the essential confidentiality in selective model there exists an *Adversary* has unneglectable advantage in winning the following game:

Phase 1 *Setup*:

*Adversary* claims an access structure  $\gamma_{ic}$  and  $\{A_{ic}\}$  is the attribute set involved.

*Challenger* runs *Setup* procedure to obtain the system parameters  $PK$  and master key  $MK$ . It sends  $PK$  to *Adversary*.

Phase 2 *Queries*:

*Adversary* can ask for the result of the following queries to *Challenger*:

*Initial private key generation* query: *Challenger* can obtain initial private key for attribute set  $S$  by running *Initial private key generation* algorithm and sends the results back to *Adversary*. Note that  $|S \cap \{A_{ic}\}| < thr_x$ ,  $thr_x$  is the threshold value of each node  $x$  in the  $\gamma_{ic}$ .

*Temporal private key generation* query: *Challenger* can obtain temporal private key for attribute set  $S$  by running *Key update* algorithms and sends the results back to *Adversary*. Note that  $|S \cap \{A_{ic}\}| < thr_x$ ,  $thr_x$  is the threshold value of each node  $x$  in the  $\gamma_{ic}$ .

The queries described in phase 2 can be asked by *Adversary* for a bounded times.

Phase 3 *Challenge*:

At the current time period  $TP_n$ , *Adversary* picks  $M_0$  and  $M_1$ , which haven't queried before. *Challenger* chooses  $\sigma \in \{0,1\}$  randomly and calculates  $CT^* = \text{Encrypt}\{PK, M_\sigma, \gamma_{ic}\}$ . Then *Challenger* sends the result back to *Adversary*.

*Adversary* outputs a value  $\sigma^*$  as a conjecture of  $\sigma$ . If  $\sigma^* = \sigma$  then *Adversary* wins the game. *Adversary* cannot ask *Challenger* for *Temporal private key generation* query for the challenging attribute set  $\{A_{ic}\}$ .

Denote  $\text{Adv}(A) = \left| \text{Pr}[\sigma^* = \sigma] - \frac{1}{2} \right|$  to be the advantage in the challenge game.

#### 4. Constructions

The concrete constructions of our KI-CPABE-KEA are described as follows:

*Setup*: Let  $G_1$  and  $G_2$  be two cyclic groups with prime order  $p$ . Denote  $g$  is the generator of  $G_1$ . Let  $\hat{e} : G_1 \times G_1 \rightarrow G_2$  be a bilinear pairing. Define a Lagrange interpolation function  $\Delta_{i,S}(x) = \prod_{j \in S, j \neq i} \frac{x-j}{i-j}$ . Define a hash function:  $H_1 : \{0,1\}^* \rightarrow G_1$ . AA randomly chooses  $y, u \in Z_p^*$ . For each attribute  $A_i$ , AA randomly chooses  $t_i \in Z_p^*$ . Key helper picks  $h \in Z_p^*$  as its master key and calculates  $g^h$ . The system master keys are  $\{t_i, \beta, u, g^y, h\}$ , system public parameters are  $\{G_1, G_2, p, g, \hat{e}, g^h, \hat{e}(g, g)^y, g^u, H_1, g^\beta\}$ .

*Initial private key generation*: At initial time period  $TP_0$ , AA generates the initial private key  $SK_{id,TP_0}$  for a user possessing attribute set  $\{A_i\}$  as follows:

$SK_{id,TP_0} = \{K = g^{\frac{y+r}{\beta}} \cdot H_1(id)^u, \forall A_i \in S, D_{i,TP_0} = g^r H_1(A_i)^{t_i} H_1(A_i, TP_0)^h, D'_i = g^{t_i}\}$   
Note that  $K, D'_i$  remain unchanged throughout the whole system lifetime, while  $D_{i,TP_n}$  updates when system enters a new time period.

*Key update*: In order to update user's private key from time period  $TP_{n-1}$  to  $TP_n$ , key helper calculates the updated key component  $UP_{i,TP_n}$  for each attribute  $i$  as  $UP_{i,TP_n} = \left( \frac{H_1(A_i, TP_n)}{H_1(A_i, TP_{n-1})} \right)^h$  and delivers it to users. Users update their temporal private keys to the latest version by calculating  $D_{i,TP_n} = D_{i,TP_{n-1}} \cdot UP_{i,TP_n}$ . The format of temporal private key at  $TP_n$  can be denoted by:

$SK_{id,TP_n} = \{K = g^{\frac{y+r}{\beta}} \cdot H_1(id)^u, \forall A_i \in S, D_{i,TP_n} = g^r H_1(A_i)^{t_i} H_1(A_i, TP_n)^h, D'_i = g^{t_i}\}$ .

*Encrypt*: Data owner randomly chooses a polynomial  $q_x$  for each node  $x$  in the access control structure  $\gamma$ . Denote  $d_x$  to be the degree of  $q_x$  and  $thr_x$  to be the threshold value node. Let  $d_x = thr_x - 1$ . For the root node data owner sets  $q_{root}(0) = s$ . For any other node (except for root node) in  $\gamma$ , let  $q_x(0) = q_{parent(x)}^{index(x)}$ . Let  $\{i\}$  to be the leaf nodes in  $\gamma$ , then the ciphertext is constructed as:

$$\begin{aligned} C_0 &= M \hat{e}(g, g)^{ys} \\ C_1 &= g^{\beta s} \\ C_2 &= g^{u \cdot s} \\ \forall i \in \gamma, C_{1,i} &= g^{q_i(0)}, C_{2,i} = H_1(A_i)^{q_i(0)}, C_{3,i} = H_1(A_i, TP_n)^{q_i(0)} \end{aligned} \quad (1)$$

Data owner uploads the ciphertext  $CT = \{C_0, C_1, C_2, C_{1,i}, C_{1,i}\}$  to cloud service provider.

*Decrypt*: Upon receiving  $CT\{C_0, C_1, C_2, C_{1,i}, C_{1,i}\}$  from cloud service provider, data receiver decrypts the ciphertext by calculating:

$$M = \frac{C_0 \cdot \hat{e}(H_1(id), C_2)}{\hat{e}(K, C_1)} \prod_{i \in \gamma} \frac{\hat{e}(D_{i,TP_n}, C_{1,i})}{\hat{e}(D'_{i,C_2,i}) \cdot \hat{e}(g^h, C_{3,i})} \quad (2)$$

Correctness proof:

For each leaf node in the  $\gamma$ , we have:

$$\begin{aligned} \frac{\hat{e}(D_{i,TP_n}, C_{1,i})}{\hat{e}(D'_i, C_{2,i}) \cdot \hat{e}(g^h, C_{3,i})} &= \frac{\hat{e}(g^r H_1(A_i)^{t_i} H_1(A_i, TP_n)^h, g^{q_i(0)})}{\hat{e}(g^{t_i}, H_1(A_i)^{q_i(0)}) \cdot \hat{e}(g^h, H_1(A_i, TP_n)^{q_i(0)})} \\ &= \hat{e}(g^r, g^{q_i(0)}) \\ &= \hat{e}(g, g)^{r q_i(0)} \end{aligned} \quad (3)$$

For each non-leaf node, let  $i = \text{index}(z)$ ,  $S_{x'} = \{\text{index}(z) : z \in S_x\}$ .

$$\begin{aligned} F_x &= \prod_{z \in S_x} F_z^{\Delta_{i,S_{x'}(0)}} = \prod_{z \in S_x} (\hat{e}(g, g)^{r q_z(0)})^{\Delta_{i,S_{x'}(0)}} \\ &= \prod_{z \in S_x} (\hat{e}(g, g)^{r q_{\text{parent}(z)}(\text{index}(z))})^{\Delta_{i,S_{x'}(0)}} \\ &= \prod_{z \in S_x} (\hat{e}(g, g)^{r q_x(i)})^{\Delta_{i,S_{x'}(0)}} \\ &= \hat{e}(g, g)^{r q_x(0)} \end{aligned} \quad (4)$$

Thus, the algorithm recovers the value of  $F_{root} = \hat{e}(g, g)^{rs}$  using Lagrange interpolation function. Then the plaintext can be recovered by:

$$\begin{aligned} M &= \frac{C_0 \cdot \hat{e}(H_1(id), C_2) \hat{e}(g, g)^{rs}}{\hat{e}(K, C_1)} \\ &= \frac{C_0 \cdot \hat{e}(H_1(id), g^{u \cdot s}) \hat{e}(g, g)^{rs}}{g^{\frac{y+r}{\beta}} H_1(id)^u, g^{\beta s}} \\ &= \frac{M \cdot \hat{e}(g, g)^{y s}}{\hat{e}(g, g)^{y s}} \\ &= M \end{aligned} \quad (5)$$

## 5. Security proof and performance analysis

### 5.1 Security proof:

Theorem: If our scheme can be broken by *Adversary* in the selective model, then a simulator can be constructed to break the DBDH hardness assumption with an unneglectable advantage.

Proof: In the challenge game, if there exists an *Adversary* which has advantage  $\varepsilon$  in attacking our KI-CPABE-KEA, there exists a simulator which is capable of breaking the DBDH hardness assumption with an advantage of  $\varepsilon/2$ .

The simulator is constructed as follows:

Phase 1 *Setup*:

Defines a global attribute set  $U = \{1, 2, \dots, i\}$ . Defines  $G_1$  and  $G_2$  be two cyclic groups of prime order  $p$ . The generator of  $G_1$  is denoted by  $g$ . Defines a bilinear pairing  $\hat{e}: G_1 \times G_1 \rightarrow G_2$ . Defines a hash function  $H_1: \{0, 1\}^* \rightarrow G$ . Randomly picks  $\mu \in \{0, 1\}$ ,  $a, b, c, z \in Z_p$ .

$$\text{Let } \begin{cases} (A, B, C, Z) = (g^a, g^b, g^c, \hat{e}(g, g)^{abc}) & \text{if } \mu = 0 \\ (A, B, C, Z) = (g^a, g^b, g^c, \hat{e}(g, g)^z) & \text{if } \mu = 1 \end{cases}$$

Simulator sets the public parameters as follows:

$$PK = \{G_1, G_2, \hat{e}, H_1, g^a, g^b, U\}.$$

The aim of simulator is to output a value  $\mu^*$  as a guess of  $\mu$ .

*Adversary* defines an attribute set  $\{A_{ic}\}$  with an encryption policy  $\gamma_{ic}$  and plays the challenge game in it. Simulator acts as *Challenger* and runs *Adversary* as a subprogram.

Phase 2: *Queries*

*Adversary* can make the following queries to simulator:

*Initial private key generation* query: *Adversary* submits queries for initial private key generation for an attribute set  $S$  (provided  $|S \cap \{A_{ic}\}| < thr_x$ ,  $thr_x$  is the threshold value of each node in the  $\gamma_{ic}$ ).

Simulator picks  $\alpha_i, \beta_i, h, u, r, j \in Z_p^*$  and sets  $SK_{id,TP_0}$  as follows:

$$SK_{id,TP_0}: \begin{cases} K = g^{\frac{ab+r}{\beta}} H_1(id)^u, \forall A_i \in S, D_{i,TP_0} = g^r H_1(A_i)^{\alpha_i} H_1(A_i, TP_0)^h, D'_i = g^{\alpha_i}, \text{ if } A_i \in A_{ic} \\ K = g^{\frac{ab+j}{\beta}} H_1(id)^u, \forall A_i \in S, D_{i,TP_0} = g^j H_1(A_i)^{\beta_i} H_1(A_i, TP_0)^h, D'_i = g^{\beta_i}, \text{ if } A_i \notin A_{ic} \end{cases}$$

Simulator sends  $SK_{id,TP_0}$  to *Adversary*.

*Temporal private key generation* query: *Adversary* submits queries for temporal private key generation for an attribute set  $S$ . Simulator responds  $SK_{id,TP_n}$  as follows:

$$SK_{id,TP_n}: \begin{cases} K = g^{\frac{ab+r}{\beta}} \cdot H_1(id)^u, \forall A_i \in S, D_{i,TP_n} = g^r H_1(A_i)^{\alpha_i} H_1(A_i, TP_n)^h, D'_i = g^{\alpha_i}, \text{ if } A_i \in A_{ic} \\ K = g^{\frac{ab+j}{\beta}} \cdot H_1(id)^u, \forall A_i \in S, D_{i,TP_n} = g^j H_1(A_i)^{\beta_i} H_1(A_i, TP_n)^h, D'_i = g^{\beta_i}, \text{ if } A_i \notin A_{ic} \end{cases}$$

*Adversary* can make a bounded queries as described above. It can be seen that the simulator is consistent with our scheme.

*Challenge*:

*Adversary* picks plaintexts  $M_0$  and  $M_1$ . Simulator chooses  $\sigma \in \{0,1\}$  and calculates  $CT_\sigma = \text{Encrypt}\{PK, M_\sigma, \gamma_{ic}\}$ . The encryption process is as follows:

Randomly chooses a polynomial  $q_x$  for each node  $x$  in the user's access control structure  $\gamma_{ic}$ . Denote  $d_x$  to be the degree of  $q_x$  and  $thr_x$  to be the threshold value of each node. Let  $d_x = thr_x - 1$ . Simulator sets  $q_{root}(0) = s$  for the root node. For any other node (except for root node) in the access tree, let  $q_x(0) = q_{parent(x)}^{index(x)}$ . Denote  $\{ic\}$  to be the leaf node of  $\gamma_{ic}$ , the ciphertext  $CT_\sigma$  is constructed as:

$$\begin{aligned} C_0 &= M_\sigma \cdot Z, C_1 = g^{\beta s}, C_2 = g^{us} \\ \forall ic \in \gamma_{ic}, C_{1,ic} &= g^{q_{ic}(0)}, C_{2,ic} = H_1(A_i)^{q_{ic}(0)}, C_{3,ic} = H_1(A_{ic}, TP_n)^{q_{ic}(0)} \\ CT_\sigma &= \{C_0, C_1, C_2, C_{1,ic}, C_{2,ic}, C_{3,ic}\} \end{aligned} \quad (7)$$

Let  $s = c$ , according to setting in *Setup* phase,  $CT_\sigma$  equals to :

$$CT_\sigma = \begin{cases} M_\sigma \hat{e}(g, g)^{abc}, C_1, C_2, C_{1,ic}, C_{2,ic}, C_{3,ic}, \text{ if } \sigma = 0 \\ M_\sigma \hat{e}(g, g)^z, C_1, C_2, C_{1,ic}, C_{2,ic}, C_{3,ic}, \text{ if } \sigma = 1 \end{cases}$$

When  $\sigma = 0$ ,  $CT_\sigma$  is a legal ciphertext in our model. *Adversary* outputs a value  $\sigma^*$  as a guess of  $\sigma$ .

If  $\sigma^* = \sigma$ , *Adversary* wins the game and the advantage can be denoted by:

$$Adv(A) = \left| Pr[\sigma^* = \sigma] - \frac{1}{2} \right|$$

Then we will discuss simulator's advantage in distinguishing the following two tuples  $\{A = g^a, B = g^b, C = g^c, \hat{e}(g, g)^{abc}\}$  and  $\{A = g^a, B = g^b, C = g^c, \hat{e}(g, g)^z\}$ .

When  $\sigma = 1$ ,  $CT_\sigma$  is a invalid ciphertext and *Adversary* cannot acquire useful information of  $\sigma$ . Under this condition the advantage can be denoted by:

$$Pr(\sigma^* \neq \sigma | \sigma = 1) = \frac{1}{2} \quad (8)$$

Since when  $\sigma^* \neq \sigma$ , the simulator outputs  $\mu = 1$ , so:



$$Pr(u^* = u | \sigma = 1) = \frac{1}{2} \quad (9)$$

When  $\sigma = 0, CT_\sigma$  is a legal ciphertext. According to the assumption, *Adversary* has an advantage  $\varepsilon$ . Under this condition the advantage can be denoted by:

$$Pr(\sigma^* = \sigma | \sigma = 0) = \frac{1}{2} + \varepsilon \quad (10)$$

Because when  $\sigma^* = \sigma$  the simulator outputs  $\mu = 0$ , so:

$$Pr(u^* = u | \sigma = 0) = \frac{1}{2} + \varepsilon \quad (11)$$

As is mentioned above, the advantage of simulator in breaking the DBDH assumption is:

$$\begin{aligned} & \frac{1}{2}Pr(u^* = u | \sigma = 0) + \frac{1}{2}Pr(u^* = u | \sigma = 1) - \frac{1}{2} \\ &= \frac{1}{2} \left( \frac{1}{2} + \varepsilon \right) + \frac{1}{2} \times \frac{1}{2} - \frac{1}{2} \\ &= \frac{\varepsilon}{2} \end{aligned} \quad (12)$$

## 5.2 Collusion resistance

In our KI-CPABE-KEA, a specific random number chosen by AA is embedded into each user's private key, consequently, it is computational infeasible combine them to decrypt a valid ciphertext. As is discussed in section 4, to decrypt a ciphertext, it is essential for the colluding attacker to recover the value of  $\hat{e}(g, g)^{ys}$ . To achieve this goal, the attacker must pair the ciphertext and the other colluding users' private keys for an attribute (assume that the attribute is not owned by attacker). However, this results in the value  $\hat{e}(g, g)^{ys} = \hat{e}(g, g)^{(y-r)s} \cdot \hat{e}(g, g)^r$  hidden by a random value  $r$ , which is uniquely distributed to users in the cloud system. This value is able to be calculated only when the corresponding private key components a user possesses satisfy with the encryption policy embedded in the ciphertext. Consequently, the target value  $\hat{e}(g, g)^{ys}$  cannot be calculated by collusion attack due to the hidden value is uniquely distributed in the private keys of each user.

## 5.3 Forward and backward security with efficient key updating

Our KI-CPABE-KEA not only achieves efficient key updating at a very low cost, but also guarantees the forward security and backward security when attribute revocation or key exposure happens. Assume that user's attribute  $A_i$  has to be revoked after the time period  $TP_n$ , then he will no longer receives the key updating information  $UP_{i, TP_{n+1}}$  from the key helper, consequently, he cannot update his temporal private key  $SK_{id, TP_n}$  to the latest version. When key exposure happens at period  $TP_n$ , the system can still maintain its security by updating users' temporal private keys to  $TP_{n+1}$  version. Without loss of generality, when a user's private key leaks during period  $TP_n$ , the system still maintains safely after  $TP_n$  since all the temporal private keys have been securely updated. This meets the requirements of both forward security and backward security. Besides, our scheme also supports random access key updating since key helper is capable of updating users' temporal private keys from any previous time periods (denote these time periods by  $TP_a$ ) to the lasted version in just one step by calculating  $D_{i, TP_n} = g^r H_1(A_i)^{\alpha_i} H_1(A_i, TP_a)^h \cdot \left( \frac{H_1(A_i, TP_n)}{H_1(A_i, TP_a)} \right)^h$ .

Then we will discuss the computation cost of key updating in our key-insulated scheme. When period  $TP_{n+1}$  arrives, key helper calculates  $UP_{i, TP_{n+1}} = \left( \frac{H_1(A_i, TP_{n+1})}{H_1(A_i, TP_n)} \right)^h$  and sends it to user. User updates his private key by calculating  $D_{i, TP_{n+1}} = D_{i, TP_n} \cdot UP_{i, TP_{n+1}}$ . During the whole key updating process, for each attribute key helper has to run 2 hash operation and 1

exponential operation, user only runs 1 multiply operation. Compared with many ABE schemes with key refreshing mechanism, our scheme has lower computation in the key updating process. Besides, the system parameters remain unchanged during the system lifetime, and the key updating process only needs the participation of key helper and user, which relieves AA from the heavy computation load of key-regeneration.

### 5.4 Traitor tracing

When a malicious user (denote  $mid$  as his unique identity and  $SK_{mid,TP_n}$  as the private key he owns) leaks his private key deliberately in the cloud system for illegal data sharing, then his identity can be exactly pinpointed by tracer. Two main methods can be adopted for traitor tracing as follows:

a. Upon receiving a legal private key  $SK_{mid,TP_n} = \{K = g^{\frac{y+r}{\beta}} \cdot H_1(mid)^u, \forall A_i \in S, D_{i,TP_n} = g^r H_1(A_i)^{t_i} H_1(A_i, TP_n)^h, D'_i = g^{t_i}\}$  from CSP, tracer firstly recovers the attribute set belonging to the malicious user from  $D'_i$  and calculates  $g^r$  as follows:

$$g^r = D_{i,TP_n} \cdot H_1(A_i)^{-t_i} H_1(A_i, TP_n)^{-h} \tag{13}$$

Then, the identity can be pinpointed by:

$$H_1(mid) = \left( K \cdot \left( g^{\frac{r}{\beta}} \cdot g^{\frac{y}{\beta}} \right)^{-1} \right)^{u^{-1}} = \left( g^{\frac{y+r}{\beta}} \cdot H_1(mid)^u \cdot \left( g^{\frac{r}{\beta}} \cdot g^{\frac{y}{\beta}} \right)^{-1} \right)^{u^{-1}} \tag{14}$$

Correctness proof:

$$\begin{aligned} g^r &= D_{i,TP_n} \cdot H_1(A_i)^{-t_i} H_1(A_i, TP_n)^{-h} \\ &= g^r H_1(A_i)^{t_i} H_1(A_i, TP_n)^h \cdot H_1(A_i)^{-t_i} H_1(A_i, TP_n)^{-h} \\ &= g^r \\ H_1(mid) &= \left( K \cdot \left( g^{\frac{r}{\beta}} \cdot g^{\frac{y}{\beta}} \right)^{-1} \right)^{u^{-1}} \\ &= \left( g^{\frac{y+r}{\beta}} \cdot H_1(mid)^u \cdot \left( g^{\frac{r}{\beta}} \cdot g^{\frac{y}{\beta}} \right)^{-1} \right)^{u^{-1}} \\ &= H_1(mid)^{uu^{-1}} \\ &= H_1(mid) \end{aligned} \tag{15}$$

b. Since user's private key is unique, if the amount of users is not huge, tracer can build a list recoding each private key with its corresponding user's identity as **Table 1** shows. When private key exposure happens, tracer searches the identifier which corresponds to the leaked private key in the list and the traitor is able to be exactly traced.

**Table 1.** List of each private key with its corresponding user's identity

User's identity	Corresponding private key
$id_1$	$SK_{id_1,TP_n}$
$id_2$	$SK_{id_2,TP_n}$
...	...
$id_n$	$SK_{id_n,TP_n}$

## 5.5 Efficiency comparison

In attribute based cryptosystem, the pairing operation and the exponential operation will consume more computation cost than other operations [17]. Thus, reducing the number of pairing operation and exponential operation will increase the efficiency of the algorithm to a large extent. In our scheme, according to the *Encrypt* algorithm, the data owner needs to run  $3n+3$  times of exponential operation (assuming the amount of attributes concerned in encryption is  $n$ ). While in decryption, each data receiver takes  $3n+2$  times of pairing operation.

With regard to the sizes of private keys and ciphertext, according to *Initial private key generation* and *Encrypt* algorithms, the size of private key for a single user is  $2n+1|p|$ , and the size of ciphertext is  $3n+2|p|$ . We denote “Exp” as exponential operation and “Pair” as pairing operation, the performance comparison results with schemes in [12] and [26] are listed in Table 2.

Table 2. Efficiency comparison

Scheme	Encryption Cost	Decryption Cost	Size of private key	Size of ciphertext	Key exposure protection
[12]	$(4n+4)$ Exp	$(4n+1)$ Pair+ $3$ Exp	$4n p $	$(4n+2) p $	No
[26]	$(5n+3)$ Exp	$(3n+1)$ Pair+ $(n+2)$ Exp	$(2n+4) p $	$(3n+3) p $	No
Ours	$(3n+3)$ Exp	$(3n+2)$ Pair+ $n$ Exp	$(2n+1) p $	$(3n+2) p $	Yes

From Table 2, it can be figured out that the overall computation cost of encryption and decryption is lower in our scheme. Besides, our scheme outperforms with respect to the sizes of private key and ciphertext. As is discussed above, the overall efficiency of our scheme is higher compared to previous schemes in [12][26].

## 6. Conclusion

We center on the demand of key exposure protection in attribute based cryptosystem and construct a key-insulated ciphertext policy attribute based encryption with key exposure accountability (KI-CPABE-KEA) for secure data protection in cloud computing. When a malicious user leaks his private key deliberately for illegal data sharing, the tracer can pinpoint his identity exactly. Besides, the key-insulation mechanism guarantees forward and backward security when key exposure happens. The higher efficiency with proved security make our KI-CPABE-KEA more appropriate for secure data sharing in cloud computing.

Our future research direction should focus on the key-insulated attribute based signature, which serves as an effective tool for data authentication in attribute based cryptosystem.

## References

- [1] Han ND, Han LZ, Tuan DM, In HP and Jo M, "A scheme for Data Confidentiality in Cloud-assisted Wireless Body Area Networks," *Information Sciences*, vol. 284, no.10, pp 157-166, Nov., 2013. [Article \(CrossRef Link\)](#)
- [2] V. Goyal, O. Pandey, A. Sahai and B. Waters, "Attribute Based Encryption for Fine-Grained Access Control of Encrypted Data," in *Proc. of ACM conference on Computer and Communications Security*, pp. 89-98, Oct.30-Nov.3, 2006. [Article \(CrossRef Link\)](#)

- [3] Waters, B., "Ciphertext policy attribute based encryption: an expressive, efficient, and provably secure realization," in *Proc. of Int. Conf. PKC 2011*, pp. 53-70, Mar. 6-9, 2011.  
[Article \(CrossRef Link\)](#)
- [4] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption," *Advances in Cryptology—EUROCRYPT 2010*, pp. 62-91, Springer, Berlin, Germany, May 30-Jun.3, 2010.  
[Article \(CrossRef Link\)](#)
- [5] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," *Advances in Cryptology—EUROCRYPT 2011*, pp.568–588, May 15-19, 2011. [Article \(CrossRef Link\)](#)
- [6] Attrapadung N, Libert B, De Panafieu E, "Expressive key-policy attribute-based encryption with constant-size ciphertexts," *Public Key Cryptography—PKC 2011*, vol. 6571 of LNCS. Springer, pp. 90-108, Mar. 6-9, 2011. [Article \(CrossRef Link\)](#)
- [7] J. Han, W. Susilo, Y. Mu, and J. Yan, "Privacy-preserving decentralized key-policy attribute-based encryption," in *Proc. of 2012 IEEE Transactions on Parallel and Distributed Systems*, vol.23, no.11, pp.2150-2162, Nov.,2012. [Article \(CrossRef Link\)](#)
- [8] Chunqiang Hu, Nan Zhang, "Body Area Network Security: A Fuzzy Attribute-Based Signcryption Scheme," *IEEE Journal on Selected Areas in Communications/SUPPLEMENT*, vol.31, no.9, pp 37-46, Sep., 2013. [Article \(CrossRef Link\)](#)
- [9] Hur J, Noh D K., "Attribute-based access control with efficient revocation in data outsourcing systems," *Transactions on Parallel and Distributed Systems, IEEE*, vol.22,no.7, pp. 1214-1221, Jul.,2011. [Article \(CrossRef Link\)](#)
- [10] M. Jason Hinek, Shaoquan Jiang, Reihaneh Safavi-Naini, "Attribute-Based Encryption with Key Cloning Protection," Available at <http://eprint.iacr.org/2008/478>
- [11] Jin Li, Kui Ren, Bo Zhu, "Privacy-Aware Attribute-Based Encryption with User Accountability," *Volume 5735 of the series Lecture Notes in Computer Science*, pp. 347-362, Sep.7-9, 2009.  
[Article \(CrossRef Link\)](#)
- [12] Fatos Xhafa, Jianglang Feng, Yinghui Zhang, "Privacy-aware attribute-based PHR sharing with user accountability in cloud computing," *Journal of Supercomputing*, vol.71, no.5:pp.1607–1619, May, 2015.  
[Article \(CrossRef Link\)](#)
- [13] YongTao Wang, KeFei Chen, Yu Long, "Accountable authority key policy attribute-based encryption," *Science China Information Sciences*, Vol. 55, Issue 7, pp. 1631-1638, Jul., 2012.  
[Article \(CrossRef Link\)](#)
- [14] LI Ming, YU Shucheng, et al., "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol.24, no.1: 131-143, Jan., 2013. [Article \(CrossRef Link\)](#)
- [15] Yu S, Wang C, Ren K, et al., "Attribute based data sharing with attribute revocation," in *Proc. of the 5th Symposium on Information, Computer and Communications Security (ACM)*, pp. 261-270, Apr.13-16,2010. [Article \(CrossRef Link\)](#)
- [16] Ximeng Liu, Hui Zhu, Jianfeng Ma, "Attribute Based Multisignature Scheme for Wireless Communications," available at <http://www.hindawi.com/journals/misy/2015/827320/>.  
[Article \(CrossRef Link\)](#)
- [17] Chen L, Cheng Z, and Smart N P., "Identity-based key agreement protocols from Pairings," *International Journal of Information security*, vol.6, no.4, pp. 213-241, Jul., 2007.  
[Article \(CrossRef Link\)](#)
- [18] J.-M. Do, Y.-J. Song, and N. Park, "Attribute Based Proxy Re-Encryption for Data Confidentiality in Cloud Computing Environments," in *Proc. of First ACIS/JNU Int'l Conf. Computers, Networks, Systems and Industrial Eng. (CNSI)*, pp. 248-251, May, 2011.  
[Article \(CrossRef Link\)](#)
- [19] Shucheng Yu, Kui Ren, Wenjing Lou, "Defending against Key Abuse Attacks in KP-ABE Enabled Broadcast Systems," *the series Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol. 19, pp.311-329, Sep. 14-18, 2009.  
[Article \(CrossRef Link\)](#)

- [20] S. Yu, K. Ren, and W. Lou, "FDAC: toward fine-grained distributed data access control in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 4, pp. 673–686, Apr., 2011. [Article \(CrossRef Link\)](#)
- [21] Dodis Y, Katz J, Xu S, Yung M. "Key-Insulated public-key cryptosystems," in *Proc. of the Eurocrypt 2002*. LNCS 2332, Berlin: Springer-Verlag, pp.65–82, Apr. 28-May 2, 2002. [Article \(CrossRef Link\)](#)
- [22] Ximeng Liu, Qi Li, Jianfeng Ma, Rui Li, Jinbo Xiong, "Provably secure unbounded multi-authority ciphertext-policy attribute-based encryption," *Security and Communication Networks*, vol. 8, no.18, pp. 4098-4109, Dec., 2015. [Article \(CrossRef Link\)](#)
- [23] Qi Li, Jianfeng Ma, Rui Li, Jinbo Xiong, Ximeng Liu, "Large universe decentralized key-policy attribute-based encryption," *Security and Communication Networks*, vol.8, no.3, pp. 501-509, Feb., 2015. [Article \(CrossRef Link\)](#)
- [24] Sahai, A, Waters, "Fuzzy identity-based encryption," in *Proc. of Int. Conf.EUROCRYPT 2005*, pp. 457-473, May 22-26, 2005. [Article \(CrossRef Link\)](#)
- [25] Hanshu Hong, Zhixin Sun, "High efficient key-insulated attribute based encryption scheme without bilinear pairing operations," *SpringerPlus*, vol.5, no.1, pp.1-12, Dec., 2016. [Article \(CrossRef Link\)](#)
- [26] Jianting Ning, Xiaolei Dong, Zhenfu Cao, "White-Box Traceable Ciphertext-Policy Attribute-Based Encryption Supporting Flexible Attributes," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 6, pp.1274-1288, Jun., 2015. [Article \(CrossRef Link\)](#)



**Dr Hanshu Hong** is a PHD candidate in Nanjing University of Posts and Telecommunications. His research area mainly includes information security, cryptology.



**Dr Zhixin Sun** is the dean of Internet of Things institute, Nanjing University of Posts and Telecommunications. He received his PHD degree in Nanjing University of Aeronautics and Astronautics, China in 1998 and worked as a post doctor in Seoul National University, South Korea between 2001 and 2002. He has published more than 50 literatures on journals worldwide. His research area includes information security, computer networks, computer science, etc.



**Dr Ximeng Liu** is a research fellow in Singapore Management University. His research is in the areas of cryptography and network security.