Secure Data Sharing Using Attribute Based Encryption

V.Balasubramani, D.S.Arun, A.S.Arun Varshan

CSE Department

T.J.S Engineering college

Abstract—Ciphertext-policy attribute-based encryption (CP- ABE) is a very promising encryption technique for secure data sharing in the context of cloud computing. Data owner is allowed to fully control the access policy associated with his data which to be shared. However, CP-ABE is limited to a potential eys of users have to be issued by a trusted key authority. Besides, most of the existing CP-ABE schemes cannot support attribute with arbitrary state. In this paper, we revisit attribute-based data sharing scheme in order to solve the key escrow issue but also improve the expressiveness of attribute, so that the resulting scheme is more friendly to cloud computing applications. We propose an improved two-party key issuing protocol that can guarantee that neither key authority nor cloud service provider can compromise the whole secret key of a user individually. Moreover, we introduce the concept of attribute with weight, being provided to enhance the expression of attribute, which can not only extend the expression from binary to arbitrary state, but also lighten the complexity of access policy. Therefore, both storage cost and encryption complexity for a ciphertext are relieved. The performance analysis and security proof show that the proposed scheme is able to achieve efficient and secure data sharing in cloud computing.

Index Terms—Secure data sharing, Attribute-based encryp- tion, Removing escrow, Weighted attribute, Cloud computing.

I.INTRODUCTION

CLOUD computing has become a research hot-spot due to its distinguished long-list advantages (e.g. convenience, high scalability). One of the most promising cloud computing applications is on-line data sharing, such as photo sharing in On-line Social Networks among more than one billion users [18], [22], [31] and on-line health record system [21], [26], [36]. A data owner (DO) is usually willing to store large amounts of data in cloud for saving the cost on local data management. Without any data protection mechanism, cloud service provider (CSP), however, can fully gain access to all data of the user. This brings a potential security risk to the user, since CSP may compromise the data for commercial benefits. Accordingly, how to securely and efficiently share

 Shulan Wang, Jianping Yu and Weixin Xie are with ATR Key Labo- ratory of National Defense Technology and College of Information En- gineering, Shenzhen University, Shenzhen, P.R. China (e-mail: wangshu- lan@email.szu.edu.cn, yujp@szu.edu.cn, wxxie@szu.edu.cn). Kaitai Liang is with Department of Computer Science, Aalto University, Finland (e-mail: kaitai.liang@aalto.fi). Joseph K. Liu is with Faculty of Information Technology, Monash Univer- sity, Australia (e-mail: joseph.liu@monash.edu). Jianyong Chen is with College of Computer and Software Engineering, Shenzhen University, Shenzhen, P.R. China (e-mail: jychen@szu.edu.cn).

user data is one of the toughest challenges in the scenario of cloud computing [1], [10], [17], [19], [20], [23], [25], [34], [38]. Ciphertext-policy attribute-based encryption (CP-ABE) [2], [4], [8], [12], [35], [39] has turned to be an important encryp- tion technology to tackle the challenge of secure data sharing. In a CP-ABE, user's secret key is described by an attribute set, and ciphertext is associated with an access structure. DO is allowed to define access structure over the universe of attributes. A user can decrypt a given ciphertext only if his/her attribute set matches the access structure over the ciphertext. Employing a CP-ABE system directly into a cloud appli- cation that may yield some open problems. Firstly, all users' secret keys need to be issued by a fully trusted key authority (KA). This brings a security risk that is known as key escrow problem. By knowing the secret key of a system user, the KA can decrypt all the user's ciphertexts, which stands in total against to the will of the user. Secondly, the expressiveness of attribute set is another concern. As far as we know, most of the existing CP-ABE schemes [2], [4], [7], [8], [12], [15], [35], [37] can only describe binary state over attribute, for example, "1 - satisfying" and "0 - not-satisfying", but not dealing with arbitrary-state attribute. In this paper, the weighted attribute is introduced to not only extend attribute expression from binary to arbitrary state, but also to simplify access policy. Thus, the storage cost and encryption cost for a ciphertext can be relieved.

We use the following example to further illustrate our approach. Suppose there is a formal structure in university, in which teachers are classified into teaching assistant, lecturer, associ- ated professor and full professor. We distribute the weight of the attribute for each type of the teachers as 1, 2, 3, and 4. Therefore, these attributes can be denoted as "Teacher: 1", "Teacher: 2", "Teacher: 3" and "Teacher: 4", respectively. In this case, they can be denoted by one attribute which has just different weights. In particular, it can be arbitrary- state attributes, such as "Teacher: teaching assistant, lecturer, associate professor, full professor".

We here assume that an access policy is represented as: T {("Lecturer" OR "Asso- ciate Professor" OR "Full Professor") AND "Male"}, and the existing CP-ABE schemes are executed on the form of access policy T. If our proposed scheme is deployed, the T can be simplified as T′ {"Teacher: 2" AND "Male"}, since the attribute "Teacher: 2" denotes the minimum level in the access policy and includes {"Teacher: 2", "Teacher: 3" "Teacher: 4"} by default. Therefore, the storage overhead of the corresponding ciphertext and the computational cost used in encryption can be reduced. These two structures areshown in Fig. 1. In addition, our method can be used to express larger attribute space than ever under the

same number of attributes. For example, if both the attribute space and weighted set include n elements, the proposed scheme can describe n2 different possibilities. In contrast, the existing CP- ABE schemes only show 2n possibilities.

A. Related Work

In 2005, Sahai and Waters [32] introduced fuzzyidentitybasedencryption(IBE),whichistheseminalworkof attribute- based encryption (ABE). After that, two variants of ABE were proposed: key-policy ABE (KP-ABE) [14] and CP-ABE [4], [8], depending on if a given policy is associated with either a ciphertext and a key. Later, many CP-ABE schemes with specific features have been presented in the literature. For example, [37] presented a novel access control scheme in cloud computing with efficient attribute and user revocation. The computational overhead is significantly eliminated from O(2N) to O(N) in user key generation by improving CP- ABE scheme, where N is the number of attributes. The size of cipher text is approximately reduced to half of original size. However, the security proof of the scheme is not fully given.

Most of the existing CP-ABE schemes require a full trusted authority with its own master secret key as input to generate and issue the secret keys of users [4], [8], [13], [14], [27], [28], [32], [35], [37]. Thus, the key escrow issue is inherent, such that the authority has the "power" to decrypt all the cipher texts of system users. Chase et al. [7] presented a distributed KP-ABE scheme to solve the key escrow problem in a multi-authority system. In this approach, all authorities, which are not colluded with each other, are participating in the key generation protocol in a distributed way, such that they cannot pool their data and link multiple attribute sets belonging to the same user. Because there is no centralized authority with master secret information, all attribute authorities should communicate with others in the system to create a user's secret key. But, a major concern of this approach is the performance degradation [6], [30]. It results in O(N2) communication overhead on both the system setup phase and any rekeying phase. It also requires each user to store O(N2) additional auxiliary key components in addition to the attribute keys, where N is the number of authorities in the system. Chow [9] later proposed an anonymous private key generation protocol for IBE where a KA can issue private key to an authenticated user without knowing the list of the user's identities. It seems
that this approach can properly be used in the context of ABE if attributes are treated as identities. However, this scheme cannot be adopted for CP-ABE, since the identity of user is a set of attributes which is not publicly unknown.

In 2013, [15] provided an improved security data sharing scheme based on the classic CP-ABE [4]. The key escrow issue is addressed by using an escrow-free key issuing protocol where the key generation center and the data storage center work together to generate secret key for user. Therefore, the computational cost in generating user's secret key increases because the protocol requires interactive computation between the both parties.

Besides, Liu et al. [27], [28] presented a fine-grained access control scheme with attribute hierarchy, where [27] and [28] are built on top of [8] and [35], respectively. In the schemes, the attributes are divided into multiple lev- els to achieve fine-grained access control for hierarchical attributes, but the attributes can only express binary state. Later, Fan et al. [13] proposed an arbitrary-state ABE to solve the issue of the dynamic membership management. In this paper, a traditional attribute is divided to two parts: attribute and its value. For example, the traditional at- tributes can be denoted as{"Doctor","Professor","Engineer"}.Attributes{Career:"Doc - tor","Professor","Engineer"}, where "Career" represents an attribute and "Doctor","Professor" and "Engineer" denote the values of the attribute "Career". Accordingly, the computation cost for attributes is more expensive than that of the traditional schemes under the same number of attributes. We note that there are some other research works on CP- ABE, such as [24], [25]. Nevertheless, they leverage different techniques to achieve data sharing. We will not compare them with our present system.

B. Our Contributions

Inspired by [37], we propose an attribute-based data sharing scheme for cloud computing applications, which is denoted as ciphertext-policy weighted ABE scheme with removing escrow (CP-WABE-RE). It successfully resolves two types of problems: key escrow and arbitrary-sate attribute expression. The contributions of our work are as follows:

• we propose an improved key issuing protocol to resolve the key escrow problem of CP-ABE in cloud computing. The protocol can prevent KA and CSP from knowing each other's master secret key so that none of them can create the whole secret keys of users individually. Thus, the fully trusted KA can be semi-trusted. Data confidentiality and privacy can be ensured.

• we present weighted attribute to improve the expression of attribute. The weighted attribute can not only express arbitrary-state attribute (instead of the traditional binary state), but also reduce the complexity of access policy. Thus the storage cost of ciphertext and computation complexity in encryption can be reduced. Besides, it can express larger attribute space than ever under the same condition. Note that the efficiency analysis will be presented in Section V.

• we conduct and implement comprehensive experiment for the proposed scheme. The simulation shows that CP- WABE-RE scheme is efficient both in terms of computation complexity and storage cost. In addition, the security of CP-WABE-RE scheme is generic group model.

II. PRELIMINARIES

A. Access Structure

Let {P1,...,Pn} be a set of parties. A collection A ⊆2 {P1,...,Pn} is monotone if ∀B,C: if B ∈ A and B ⊆ C then C ∈ A. An access structure (respectively, monotone access structure) is a collection (respectively, monotone col- lection) A of non-empty subsets of {P1,...,Pn}, i.e., A ⊆ 2{P1,...,Pn}\{∅}. The sets in A are called authorization sets. Otherwise, the sets are called unauthorization sets. In our scheme, the role of the parties is taken by the attributes. Thus, A is going to include the authorized sets of attributes. Generally, unless stated in another way, the scheme uses an access structure which is a monotone access structure.

B. Bilinear Mapping

Let G0 and GT be two multiplicative cyclic groups of prime order p. The generator of G0 is g. A bilinear mapping ˆ e : G0 ×G0 →GT satisfies the following properties:

• Bilinearity: For any u,v ∈ G0 and a,b ∈ Zp, it has ˆ e(ua,vb) = ˆ e(u,v)ab.

• Non-degeneracy: There exists u,v ∈ G0 such that ˆ e(u,v)≠ 1.

• Computability: For all u,v ∈ G0, there is an efficient algorithm to compute ˆ e(u,v).

C. Weighted Access Tree

Let T be a weighted access tree, where root node of the tree is R. To facilitate description of the access tree, several functions and terms are defined as follows.

• x denotes a node of tree T. If x is a leaf node, it denotes an attribute with weight. If x is a non-leaf node, it denotes a threshold gate, such as "AND", "OR" and "n- of-m (n<m)". For example, the nodes C and A denote a threshold gate and an attribute respectively in Fig. 2. • numx denotes the number of x's children in T. For example, numR = 2 in Fig. 2.

• kx denotes threshold value of node x, where 0 < kx ≤ numx. When kx = 1 and x is a non-leaf node, it is an OR gate. When kx = numx and x is a non-leaf node, it is an AND gate. In particular, if x is a leaf node, kx = 1. For example, kR = 1 and kC = 2 denote an OR gate and an AND gate respectively in Fig. 2. • parent(x) represents the parent of the node x in T. For example, parent(A) = C in Fig. 2.

• att(x) denotes an attribute associated with the leaf node x in T.

• index(x) returns an unique value associated with the node x, where the value is assigned to x for a given key in an arbitrary manner. • Tx denotes the sub-tree of T rooted at the node x. If a set of weighted attribute S satisfies the access treeTx, we denote it as Tx(S) = 1. Tx(S) is recursively computed as follows. If x is a non-leaf node,Tx(S) returns 1 if and only if at least kx children return 1. If x is a leaf node, thenTx(S) returns 1 if and only if the weight of attribute ωx from S must be greater than or equal to the weight of the leaf node. That is weight(ωx) ≥ weight(att(x)). In addition, Morillo et al. [29] proved that every weighted value of the threshold access structure can be defined as a natural number. Unless stated otherwise, the value of weight is a natural number in this paper. In Fig. 2, the access policy is denoted as: {("Teacher:1" And "Seniority:2") OR "Teacher:3"}. If one possesses attributes ("Teacher", "Senior- ity") with weight ("1", "2"), he can satisfy the tree in Fig. 2; If the other one who possesses attribute ("Teacher") with weight ("4"), he can also satisfy the access tree.

III. SYSTEM MODEL

As illustrated in Fig. 3 and Fig. 4, the system model and framework of CP-WABE-RE scheme in cloud computing are given, where the system consists of four types of entities: KA, CSP, DO and Users. In addition, we provide the detailed definition of CP-WABE-RE scheme.

Key Authority (KA). It is a semi-trusted entity in cloud system. Namely, KA is honest-but-curious, which can honestly perform the assigned tasks and return correct results. However, it will collect as many sensitive contents as possible. In cloud

system, the entity is responsible for the users' enrollment. Meanwhile, it not only generates most part of system param- eter, but also creates most part of secret key for each user. Cloud Service Provider (CSP). It is the manager of cloud servers and also a semi-trusted entity which provides many services such as data storage, computation and transmission. To solve the key escrow problem, it generates both parts of system parameter and secret key for each user. Data Owners (DO). They are owners of files to be stored in cloud system. They are in charge of defining access structure and executing data encryption operation. They also upload the generated ciphertext to CSP. Users. They want to access ciphertext stored in cloud system. They download the ciphertext and execute the cor- responding decryption operation.

Definition 1. (CP-WABE-RE): The proposed scheme contains the following four phases: Phase 1 : System Initialization. This phase includes both algorithms: KA.Setup and CSP.Setup. (1) KA.Setup(1κ) → (PP1,MSK1). It is executed by KA. The probabilistic operation inputs a security parameter κ. It returns a public parameter PP1 and a master secret key MSK1. (2) CSP.Setup(1κ) → (PP2,MSK2). This algorithm is run by CSP. It inputs a security parameter κ and generates PP2 and MSK2. The public parameter and master secret key of system are denoted as PP = {PP1,PP2} and MSK = {MSK1,MSK2}, where MSK1 and MSK2 are stored by KA and CSP, respec- tively. Phase 2 : Data Encryption. To improve efficiency of encryption, DO first encrypts file M with content key ck by using simple symmetric encryption algorithm, where file ciphertext is denoted as Eck(M). Then, the content key ck is encrypted by the following operation. DO.Encrypt(PP,ck,A) → (CT). DO inputs PP, ck, and an access policy A. It encrypts ck and outputs content key ciphertext CT which implicitly contains A. Then, DO delivers Eck(M) and CT to CSP. Phase 3 : User Key Generation. This phase consists of KA.KeyGen and CSP.KeyGen.

(1) KA.KeyGen(MSK1,S) → (SK1). KA inputs MSK1 and a set of weighted attributes S. It creates secret key SK1 described by S. (2) In CSP.KeyGen, we propose an improved two-party key issuing protocol to remove escrow. KA and CSP perform the improved protocol with master secret keys of their own. Thus, none of them can create the whole set of secret keys of users individually. Meanwhile, we assume that KA does not collude with CSP since they are honest as in [16] (otherwise, they can obtain the secret keys of each user by sharing their master secret keys). CSP.KeyGen(MSK2) → (SK2). CSP inputs MSK2 and the required information. It produces secret key SK2 by executing the following key issuing protocol. • KeyComKA↔CSP(MSK1,IDt,r,MSK2) → (SK2). It is an interactive algorithm between KA and CSP. KA inputs MSK1, a user identity IDt and a personalized secret r. CSP inputs MSK2 and IDt. At last, only CSP generates a personalized key component SK2 for the corresponding user. Then, the user constructs the whole secret key SK with the key components separately receiving from KA and CSP, i.e. SK = {SK1,SK2}. Phase 4 : Data Decryption. This phase contains both algorithms: Users.Decrypt and Data.Decrypt. User first downloads file ciphertext Eck(M) and content key ciphertext CT from CSP. If he satisfies conditions, he can get content key ck by calling Users.Decrypt algorithm. Then, he uses ck to further decrypt file M by using Data.Decrypt operation. (1) Users.Decrypt(PP,SK,CT) → (ck). User inputsPP , SK described by S, and CT which includes access policy A. Only when the weighted attribute set S matches the access policy A, the content key ck is obtained. (2)

Data.Decrypt(Eck(M),ck) → (M). User inputsE ck(M) and ck. Based on symmetric decryption algorithm, it outputs file M.

IV. THE PROPOSED CP-WABE-RE SCHEME

In this section, we present the construction of CP-WABE- RE system, including five procedures: system initialization, new file creation (data encryption), new user authorization (user key generation), data file access (data decryption), and data file deletion. In addition, the revocation scheme of [15] can be directly used in our proposed scheme. The reason is described as below. The revocation scheme is performed in the phase of data encryption. And the removing escrow is operated in the phase of user key generation. Therefore, in [15], the modification of removing escrow does not affect the use of revocation scheme since they are run in different phases.

A. System Initialization Let G0 be a bilinear group of prime order p, and g be a generator of G0. Let ˆ e : G0 ×G0 → GT denote the bilinear map. Let H : {0,1}∗ → G0 be a hash function. For anyi ∈ Zp and a set S = {s1,s2,...,sm ∈ Zp}, the Lagrange coefficient Δi,S =∏l∈S,l≠i(x − l)/(i − l). In addition, an universe of attribute set A = {a1,...,an}and a set of weights

W = {ω1,...,ωn}(ω1 ≤ ... ≤ ωn) are defined. Thus the system contains n2 weighted attributes which are ˜ A = {a1 :ω 1,...,a1 : ωn,...,an : ω1,...,an : ωn}, where the higher hierarchy of attributes is used, the bigger weighted value is distributed. (1) KA.Setup(1κ). KA runs the algorithm which inputs a security parameter κ. Then, KA chooses random α1,β ∈ Zp and computes h = gβ and u1 = ˆ e(g,g)α1. Lastly, it obtains PP1 and MSK1 as the formula (1): PP1 = {G0,g,h,u1}, MSK1 = {α1,β} (1) (2) CSP.Setup(1κ). CSP executes the operation which inputs a security parameter κ. Based on the κ, CSP chooses a random number α2 ∈Zp and calculates u2 = ˆ e(g,g)α2. Then, it sets PP2 and MSK2 as the formula (2): PP2 = {u2}, MSK2 = {α2} (2) Finally, the public parameter and master secret key of sys- tem are denoted as PP = {G0,g,h,u = u1 ·u2 = ˆ e(g,g)α},where α = α1 + α2, and MSK = {{α1,β},{α2}}. B. New File Creation (Data Encryption) Before file M is uploaded to CSP, DO processes the file with the following steps: (1) DO picks a unique ID for file M. (2) It encrypts M with content key ck by using symmetric encryption method, where ck is chosen in a key space. The file ciphertext is denoted as Eck(M), where Eck denotes a symmetric encryption operation with the key ck. (3) It defines an access structure T and encrypts the ck by running the improved encryption operation. Then, content key ciphertext CT is returned. DO.Encrypt(PP,ck,T). The improved algorithm is ex- ecuted by DO which inputs PP, ck and T. It outputs CT. Firstly, a polynomial qx is selected for each node x (includ- ing the leaf nodes) in T. From the root node R, the node's information of qx is randomly selected from top to bottom manner. For each node x in T, degree of the polynomial dx is set to kx −1, where kx is the threshold value. Then, beginning from the root node R, DO sets qR(0) = s(s ∈ Zp), where s is randomly selected. And DO ran- domly selects dR other points of the polynomial qR to define it completely. For each non-root node x, it sets qx(0) = qparent(x)(index(x)) and randomly chooses dx other points to completely define qx. Meanwhile, each leaf node denotes an attribute with weight. In the access tree T, let Y be the set of leaf nodes, andω i be the minimum weight of each leaf node, which is set by DO. Then DO computes CT as the formula (3). Finally, DO sends the integrated ciphertext {ID,CT,Eck(M)} to CSP. CT =□ ¯ □ ¯ □ ¯ □ ¯ □ T, ˜ C = ck·ˆ e(g,g)α·s,C = gs □ □ ¬ □ □ ¯ ∀y ∈ Y,i ∈ [1,n] : Cy = hqy(0) ·H(att(y))−ωis, ∀j ∈ (i,n],Cy,j = H(att(y))−(ωj−ωi)s □ □ ¯ □ □ ¯ □ □ ¯ □ □ ¯ □ □ ¯ □ ¯

(3)

To better understand, let us take the Fig. 2 for example here. DO sets the polynomial qR(x) with degree 0 as qR(x) = s

for node R. At the same time, the polynomial qC(x) is set as qC(x) = s+crx with degree 1 where qR(u) = qC(0) = s and cr is randomly chosen in Z∗ p from node C. In addition, for leaf nodes A, B and D, they are respectively set as: qA(0) = qC(x) = s + crx, qB(0) = qC(y) = s + cry and qD(0) = qR(z) = s.

C. New User Authorization (User Key Generation) When a user wants to join the cloud system,KA first accepts the user's enrollment. If he is legal, KA authenticates and assigns a set of weighted attributes S to the user in accordance with his identity or role. Then, KA and CSP cooperate each other and generate secret key SK for the user. The phase consists of CSP.KeyGen and KA.KeyGen. (1) CSP.KeyGen. We provide an improved key issuing protocol between KA and CSP to execute the work of CSP. KeyComKA↔CSP(MSK1,IDt,r,MSK2). Assume that user t needs a secret key. Firstly, KA randomly chooses a number r ∈ Zp for the user. Then, KA and CSP perform a secure two-party computation (2PC) protocol, where KA inputs MSK1 = {α1,β}andCSPinputs MSK2 = {α2}.After the execution of the 2PC protocol, CSP gets x = (α1 + α2)β mod p. This can be done via a general 2PC protocol for a simple arithmetic computation [7], [9], [15]. Alternatively, we can do this more efficiently by using the scheme in [3]. Note that during the 2PC protocol, KA knows nothing about α2 while CSP knows nothing about {α1,β}. After the 2PC protocol, KA and CSP engage the following interactive protocol in order to generate the personalized key component for the user: 1) CSP selects a random number ρ1 ∈ Zp. It calculatesX 1 = gx/ρ1 = g(α1+α2)β/ρ1 and transmits {X1,PoK(ρ1,x)} to KA, where PoK represents a proof of knowledge of the secret values used in the computation. It can be efficiently realized, e.g. via Schnorr protocol. 2) KA chooses θ ∈ Zp and computes Y1 = Xθ/β 1 = g(α1+α2)θ/ρ1 and Y2 = hrθ. It sends {Y1,Y2,PoK(θ,β,r)} to CSP. 3) CSP randomly selects ρ2 ∈ Zp and computes X2 =(Y ρ1 1 Y2)ρ2 = (g(α1+α2)θhrθ)ρ2. It transfers {X2,PoK(ρ2)} to KA. 4) KA calculates Y3 = X1/θ 2 = (g(α1+α2)hr)ρ2 and sends {Y3,PoK(θ)} to CSP. 5) CSP calculates D = Y 1/ρ2 3 = g(α1+α2)hr = gαhr and sends a personalized key component SK2 = {D = gαhr} to the corresponding user t. Fig. 5 provides a direct description for the above protocol. Here the first step denotes a 2PC protocol which inputs {α1,β} from KA and {α2} from CSP, and returns x = (α1 + α2)β mod p to CSP. (2) KA.KeyGen(MSK1,r,S). KA executes the algorithm which inputs MSK1, a number r ∈ Zp which has been randomly chosen in CSP.KeyGen, and a set of weighted attributes S. Then, for each weighted attribute j ∈ S, it possesses weighted value ωj(ωj ∈ W). Finally, it computes SK1 described by S as the formula (4).

5. The improved two-party key issuing protocol. "2PC" denotes a general two-party computation protocol. "PoK" denotes a proof of knowledge of the secret values used in the computation.

SK1 = {L = gr,∀j ∈ S : Dj = H(j)rωj} (4) So, user t can construct the whole secret key SK by using the key

components separately receiving from the two entities. It is described as the formula (5).

SK = {D = gαhr,L = gr,∀j ∈ S : Dj = H(j)rωj} (5) D. Data File Access (Data Decryption) In cloud system, legal users can freely query the ciphertext. When a user requests CSP to access a ciphertext, it transmits the corresponding ciphertext {ID,CT,Eck(M)} to the user. The user can obtain content key ck by calling the improved Users.Decrypt algorithm. Then, he uses ck to further decrypt the file M using Data.Decrypt operation. (1) Users.Decrypt(PP,CT,SK). User inputs PP, CT, and SK described by S. If the weighted attributes S that the user possesses satisfy access policy T, the user can obtain content key ck. The operation is a recursive algorithm which is defined as below. 1) If x is a leaf node. Let k = att(x), ωk be the weighted value of the user's node x and ωi be the weighted value of the access policy T's node x. If k / ∈ S or k ∈ S,ωi > ωk,we note DecryptNode(CT,SK,x) = ⊥. If k ∈ S andω i = ωk, we compute DecryptNode(CT,SK,x)1 as the formula (6). If k ∈ S, ωi < ωk and ωk = ωj, we compute DecryptNode(CT,SK,x)2 as the formula (7).

DecryptNode(CT,SK,x)1 = ˆ e(Cx,L)·ˆ e(C,Dk) = ˆ e(hqx(0) ·H(att(x))−ωis,gr)·ˆ e(gs,H(k)rωk) = ˆ e(gβqx(0),gr)·ˆ e(H(k)−ωis,gr)·ˆ e(gs,H(k)rωk) = ˆ e(g,g)rβqx(0) (if ωk = ωi)

(6)

DecryptNode(CT,SK,x)2 = ˆ e(Cx ·Cx,j,L)·ˆ e(C,Dk) = ˆ e(hqx(0)H(att(x))−ωisH(att(x))−(ωj−ωi)s,gr) ·ˆ e(gs,H(k)rωk) = ˆ e(gβqx(0) ·H(k)−ωjs,gr)·ˆ e(gs,H(k)rωk) = ˆ e(gβqx(0),gr)·ˆ e(H(k)−ωjs,gr)·ˆ e(gs,H(k)rωk) = ˆ e(g,g)rβqx(0) (if ωk = ωj > ωi)

(7)

2) If x is a non-leaf node, DecryptNode(CT,SK,x) is defined: for all nodes z that are children of x, it runs DecryptNode(CT,SK,z) and stores the output as Fz. Let Sx be an arbitrary kx-sized set of child nodes z such that Fz≠ ⊥. If the nodes don't exist, Fz = ⊥. If not, Fx is computed as the formula (8), where k = index(z), and S′ x = {index(z) : z ∈ Sx}. Fx = ∏ z∈Sx F Δ k,S′ x(0) z = ∏ z∈Sx (ˆ e(g,g)r·βqz(0))Δk,S′ x(0) = ∏ z∈Sx (ˆ e(g,g)r·βqparent(z)(index(z)))Δk,S′ x(0) = ∏ z∈Sx (ˆ e(g,g)r·βqx(k))Δk,S′ x(0) = ˆ e(g,g)r·βqx(0) (8) Then, we define the decryption algorithm by calling DecryptNode(CT,SK,x)1 or DecryptNode(CT,SK,x)2 on the root node R of the access tree T. If the T is satisfied by S, we define A = DecryptNode1or2(CT,SK,R) = ˆ e(g,g)rβqR(0) = ˆ e(g,g)rβs. Thus, the user can gain ck with the formula (9). e C/(ˆ e(C,D)/A) = e C/(ˆ e(gs,gα ·hr)/ˆ e(g,g)rβs) = ck·ˆ e(g,g)αs/ˆ e(g,g)αs = ck (9) (2) Data.Decrypt(Eck(M),ck). User inputs file cipher- text Eck(M) and content key ck. Based on symmetric de- cryption algorithm, i.e., DES or AES, the file M can be decrypted as the formula (10), where Dck denotes a symmetric decryption operation with the key ck.

Dck[Eck(M)] = M (10)

E. Data File Deletion Here, we show that data file deletion can perform both discretionary deletion and mandatory blocking. Discretionary Deletion. All of the legal data owners can freely delete ciphertext in cloud system. Assume that a DO wants to delete an encrypted file, the procedures of algorithm between DO and CSP are described as below, where the algorithm can adopt any secure signature scheme such as BLS short signature scheme [5] as the underlying primitive to achieve. (1) DO sends a request to CSP, which includes file's ID and its signature on the ID. (2) CSP verifies these request information. If validation, CSP deletes the corresponding ciphertext. Mandatory Blocking. To provide legitimate aspect of file sharing, a new function, i.e., mandatory blocking, is added to the proposed system. The steps are described as below.

(1) When accessing a file, user needs to evaluate how well the file is accessed, such as shopping online and teaching online.
(2) CSP synthesizes these assessments for each file. If some files are not consistent with the evaluation standards, those files would be mandatory blocked by CSP. Meanwhile, DO will receive the private messages explaining the reason.

V. PERFORMANCE ANALYSIS

In this section, we analyze and compare the efficiency of the proposed scheme with the schemes [15], [37] and [13] in theoretical and experimental aspects.

A. Theoretical Analysis

1) Key Escrow and Weighted Attribute: Table I shows the problem of key escrow, feature of weighted attribute and application in cloud computing for each scheme. The key escrow in CP-WABE-RE scheme can be removed by using an improved key issuing protocol for cloud computing. [15] uses escrow-free key issuing protocol to solve the issue. On the contrary, both [37] and [13] don't solve the problem of key escrow. In addition, the weighted attribute in CP-WABE-RE scheme can not only support arbitrary-state attribute in-stead of the traditional binary state, but also simplify access policy associated with a ciphertext as opposed to [15] [37]. Unfortunately, [13] can only express arbitrary-state attribute, and cannot simplify the access structure. In Table I, we can find that only CP-WABE-RE scheme can simultaneously support all the three functions. [15] solves the problem of key escrow so it can satisfy environment of cloud system as ours. However, both [37] and [13] cannot remove key escrow. Thus the both schemes cannot be directly applied in cloud computing.

TABLE I FEATURE COMPARISONS

Scheme	Key Escrow	Weighted Attribute	Cloud System
CP-WABE-RE	No	Yes	Yes
[15]	No	No	Yes
[37]	Yes	No	No
[13]	Yes	Yes	No

2) Efficiency: In Table II and Table III, we compare ef- ficiency of the above four schemes on storage overhead and computationcostintheory,wheretheusedsymbolsaredefined in Table IV. To simplify the comparisons, access structure, data re- encryption of [15] [37], and dynamic membership manage- ment (that is, user joining, leaving, and attribute updating) of [13] are not included in the following analysis. In addition, the cost of transmission isn't involved when implementing the interactive protocols in both [15] and our proposed scheme.

TABLE IV NOTATIONS FOR EFFICIENCY COMPARISONS

Notation	Definition
Gi	exponentiation or multiplication in group (i = 0,T)
Cˆ e ˆ	e operation, ˆ e denotes bilinear pairing Zp Group
{0,1,...,p−1}	under multiplication modulo p S Least interior
nodes satisfying an access structure	
AC	Attributes appeared in ciphertext CT Au Attributes of user u
ωi	Maximum weight of attribute i in system
ωi1	Weight of attribute i in ciphertext CT

n Number of attributes in system
k Number of users in system L∗ Bit-Length of element in * |∗| Number of elements in *

In Table II, the schemes are compared in terms of CT size, SK size, PP size and MSK size. CT size represents the storage overhead in cloud computing and also implies the communication cost from DO to CSP, or from CSP to users. SK size denotes the required storage cost for each user. PP and MSK sizes represent the storage overhead of KA and CSP in terms of public parameter and master secret key. As shown in Table II, when ωi = ωi1, all attributes possess equal weights in CP-WABE-RE scheme. Thus our scheme is equivalent to [15] and [37]. Meanwhile, CT size is reduced as (|AC|+ 1)LG0 + LGT in CP-WABE-RE scheme, which is equal to [37]'s. Comparing with [15] and [13], the CT size in our proposed scheme and [37] is reduced by nearly half. When ωi≠ ωi1, CP-WABE-RE scheme can use an attribute to express (ωi−ωi1 +1) attributes which have different weights. Therefore, it requires smaller storage cost in CT than the others. Moreover, we can find that the SK size in CP-WABE-RE scheme is equal to [37]'s, which is smaller than [15]'s and [13]'s. Furthermore, when |Au|→∞, the storage overhead in our scheme is reduced by nearly half comparing to [15]'s. And the storage cost in our scheme is decreased nearly by 66.67% comparing to [13]'s in theory. In addition, we can also observe that the PP size is equal among [15], [37] and CP-WABE-RE scheme. And the size of PP in [13] is the longest since it is related to the number of system attributes n and the number of system users k. About the size of MSK, we can find that the parameter in CP-WABE-RE scheme doesn't appear to be much different from the others. In Table III, we evaluate the computation cost of encryption, decryption and user key generation. In the phase of new file creation (data encryption), the computation cost in CP-WABE- RE scheme can be reduced as (2|AC|+ 1)G0 + 2GT when

TABLE II EFFICIENCY COMPARISONS: STORAGE COST

Scheme Size of CT Size of SK Size of PP Size of MSK CP-WABE-RE [Σ|AC| i=1 (ωi −ωi1 + 1) + 1]LG0 + LGT (|Au|+ 2)LG0 3LG0 + LGT 3LZp [15] (2|AC|+ 1)LG0 + LGT (2|Au|+ 1)LG0 3LG0 + LGT LZp + LG0 [37] (|AC|+ 1)LG0 + LGT (|Au|+ 2)LG0 3LG0 + LGT LG0 [13] 2(|AC|+ 1)LG0 + 2LGT (3|Au|+ 1)LG0 (n + 2)LG0 + 2LGT + knLZp LG0

TABLE III EFFICIENCY COMPARISONS: COMPUTATION COST

Scheme New File Creation (Data Encryption) Data File Access (Data Decryption) New User Authorization (User Key Generation) CP-WABE-RE {[Σ|AC| i=1 (ωi −ωi1 + 2)] + 1}G0 + 2GT (2|Au|+ 1)Cˆ e + (2|S|+ 2)GT (|Au|+ 9)G0 [15] (2|AC|+ 1)G0 + 2GT (2|Au|+ 1)Cˆ e + (2|S|+ 2)GT (2|Au|+ 4)G0 [37] (2|AC|+ 1)G0 + 2GT (2|Au|+ 1)Cˆ e + (2|S|+ 2)GT (|Au|+ 3)G0 [13] (2|AC|+ 2)G0 + 3GT (3|Au|+ 1)Cˆ e + (2|S|+ 3)GT (4|Au|+ 2)G0

ωi = ωi1, which is roughly equal to [15]'s, [37]'s and [13]'s. Similar to Table II, when ωi≠ ωi1, CP-WABE-RE scheme computes an attribute to represent multiple attributes which possess different weights. Meanwhile, it can simplify access structure associated with a ciphertext. However, the scheme [13] doesn't possess the feature of our scheme, i.e., without expressing arbitrary-state attribute. So, when ωi≠ ωi1, the encryption cost in CP-WABE-RE scheme is saved. In the phase of file access (data decryption), the length of parameter is equal among [15], [37] and CP-WABE-RE scheme. And the computation cost on decryption in [13] is larger than the others. In addition, in the phase of new user authorization (user key generation), our proposed scheme only consumes additional 6G0 of computation cost in solving key escrow issue, comparing with that in [37]. Meanwhile, the computation cost on key generation in CP-WABE-RE scheme is smaller than [15]'s and [13]'s. Furthermore, when |Au| → ∞, the computation cost in ours is decreased nearly by 50% in theory than [15]'s, where the cost for transmission isn't involved in both the two schemes. At the same time, the cost in ours is reduced by nearly 75% comparing to [13]'s in theory.

B. Experimental Analysis Now, to validate theoretical analysis proposed in previous subsection, we execute CP-WABE-RE scheme by using the cpabe toolkit and the Java Pairing-Based Cryptography library (JPBC) [11]. Meanwhile, we also simulate the schemes in [15], [37] and [13] at the same condition. The following experiments are conducted using Java on the system with Intel(R) Core(TM) i5-4590 CPU at 3.30 GHz and 8.00GB RAM running Windows 7. To achieve a 80-bit security level, the experiments use a 160-bit elliptic curve group based on
the supersingular curve y2 = x3 + x over a 512-bit finite field. In addition, all the simulation results are the mean of 10 trials. The units of storage cost and time are Kilobyte (KB) and second (s). 1) Simulation Analysis of Key Escrow: The storage over- head and computation cost of user secret key are compared as plotted in Fig. 6. The number of weighted attributes used in this simulation is N = {10,20,30,40,50}. Fig. 6(a) and Fig. 6(b) intuitively show the experimental results. We find that the storage overhead of user secret key in ours is the same as [37]'s, and it is smaller than [15]'s and [13]'s under the same number of attributes. About the computation cost of secret key, the value of our scheme is larger than [37]'s, where the difference is 6G0 according to the Table III. At the same time, the parameter in CP- WABE-RE scheme is smaller than [15]'s and [13]'s at the same condition. We also observe that all experimental results are gradually increasing and approximately follow a linear relationship with the number of weighted attributes. Therefore, with a small error tolerance, we estimate their limit values, where the mathematical expressions are computed by using the mean algorithm. When N →∞, the limit value of space saving in CP-WABE-RE scheme is approximately equal to 48.39% comparing to [15]'s. The cost is reduced by nearly half in theory which is consistent with the above efficiency analysis. Comparing with both our scheme and [13], the saved storage cost is approximately 64.47% which matches the corresponding limit value in theory. In addition, when N → ∞, comparing with CP-WABE-RE scheme and [15], the maximum of improved efficiency in computation cost approaches to 23.04%. Comparing with our scheme and [13], the reduced computation cost is approximate to 64.88%. 6(b). However, in Table III, if |Au|→∞, the corresponding computation costs can be reduced to 50% and 75% in theory, where the computation cost for transmission isn't involved. Remarkably, the cost comes from the difference between theoretical value and experimental result. 2) Simulation Analysis of Weighted Attribute: Next, we measure and analyze the storage overhead and computation cost for encrypting (by a DO) data, where the number of attributes in access policy is N = {10,20,30,40,50}. It should be noted that the CP-WABE-RE scheme is equivalent to the schemes in [15] and [37] when all attributes possess equal weights (ωi = ωi1), where it has been analyzed in Table II and Table III. For simplicity, we have omitted the simulation when ωi = ωi1. To show the advantage of the weighted

attribute here, in CP-WABE-RE scheme, the maximum value of each weighted attribute is set as 5, and the lowest value of each weighted attribute is chosen to be encrypted. We implement [15], [37], [13] and our proposed scheme under the equivalent access policy encrypted in ciphertext. Fig. 7 shows the simulation results.

Fig. 7(a) plots the relationship between the storage overhead of ciphertext and the number of weighted attributes in access policy. Fig. 7(b) shows encryption time of ciphertext versus the number of weighted attributes. When ωi≠ ωi1 (here we assume that an attribute can be represented 5 attributes which possess different weights), we find that CP-WABE-RE scheme requires less storage cost and encryption time than the others.

We also observe that all results approximately follow a linear relationship with the number of weighted attributes in access tree. Similar as the analysis of Fig. 6, we estimate the limit values with a small error tolerance, where the mathematical expressions are computed by using the mean algorithm. For example, in Fig. 7(a), when N → ∞, comparing with CP- WABE-RE scheme and [15], the limit value of space saving is approximately equal to 52.60%. Similarly, the reduced storage cost in ciphertext is 11.77% comparing our scheme to [37]. And comparing with [13], our proposed scheme can save storage cost approximate to 51.71%. In addition, the reduced storage cost in [37] approaches to 46.28% comparing to [15]. In Fig. 7(b), when N →∞, CP-WABE-RE scheme can save computation cost approximate to 61.68% comparing with [15], [37] and [13], where the computation cost associated with a ciphertext in [15] is approximately equal to [37]'s and [13]'s. It indicates that the results are consistent with the theoretical analysis presented in previous subsection.

VI. SECURITY PROOF

We first present the chosen plaintext attacks (CPA) security proof of our CP-WABE-RE scheme. The security game is identical to those of traditional (fully) CP-ABE systems. We state here the definition of an adaptive CP-ABE security game for the completeness of the security analysis. 1) System Initi. The challenger runs the operations of KA.Setup and CSP.Setup of CP-WABE-RE scheme and sends public parameter PP to the adversary A.2) Phase 1. For the attribute sets S1,...,Sq1(∀i ∈ [1,...,q1]) chosen by A, he can repeatedly ask C for the secret key SK. Meanwhile, the challenger answers the secret key SK by running the algorithms of CSP.KeyGen and KA.KeyGen. 3) Challenge. A submits two equal length messages M0,M1 ∈ GT and an access tree A to the challenger, where there should not be any secret key issued to A such that the key satisfies A. The challenger randomly picks a bit μ ∈{0,1}and encrypts Mμ with A by using the algorithm DO.Encrypt. 4) Phase 2. Same as the Phase 1 but with the restriction that the querying key cannot satisfy A. 5) Guess. A outputs a guess ˆ μ of μ. In this game, A canwin the game which is defined as |Pr[ˆ μ = μ]−(1/2)|.

Definition2. The proposedschemeis said to besecureagainst CPA if no probabilistic polynomial-time adversaries have non- negligible advantage in the above game. We use the generic bilinear group model and the random oracle model to prove that no adversary can break the CPA security of our scheme with non-negligible probability. In other words, our security is reduced to mathematical properties of elliptic curve groups as well as security of target collision resistance hash function. We note that our security proof technique follows that of [4]. Consider two random encodings ξ0, ξ1 of an additive group Fp, which is injective maps ξ0,ξ1 : Fp → {0,1}m, where m > 3log(p). We set G0 = {ξ0(x) : x ∈ Fp} and GT = {ξ1(x) : x ∈Fp}. In the security game, the simulator is given a random oracle for simulating hash function, and oracles in groups G0,GT and bilinear map ˆ e : G0 × G0 → GT for computation queries. We are also given a random oracle to represent the hash function H. And we refer toG0 as a generic bilinear group. Below, we give a lower bound on the advantage of a generic adversary in breaking the security of our scheme.

Theorem1. ForanyadversaryA,let q beaboundonthetotal number of group elements which A receives from queries to the oracles for the hash function, groups G0, GT, the bilinear map ˆ e and from its interaction with the security game, in which G0 is bilinear group of prime order p with generator g. We have that the advantage of A in the game is O(q2/p).

Proof. In the challenge phase of a CP-ABE game, the simulator will construct either M0ˆ e(g,g)αs or M1ˆ e(g,g)αs as the component ˜ C. Here, we consider a modified game where ˜ C is either ˆ e(g,g)αs or ˆ e(g,g)θ, and θ ∈R Fp. Now, theadversary is required to tell if ˜ C = ˆ e(g,g)αs or ˆ e(g,g)θ. It is not difficult to see that the modified game can be regarded as a hybrid argument in which the adversary is asked to tell ˆ e(g,g)θ from M0ˆ e(g,g)αs, and ˆ e(g,g)θ from M1ˆ e(g,g)αs. Accordingly, an adversary in the CP-ABE game with advantage ϵ is transformed into an adversaryAin the modified game with advantage at least ϵ 2. Below we let g = ξ0(1), gx = ξ0(x) and ˆ e(g,g)x = ξ1(x). 1) System Initi. The simulator chooses α1,α2,β ∈R Fp, and next sets h = gβ, u1 = ˆ e(g,g)α1, v1 = gα1, u2 = ˆ e(g,g)α2, v2 = gα2 and α = α1 + α2. It further sends the PP = {g,h,u = u1u2} to A.2) Hash Queries. If A issues a hash query on an attribute att(y), the simulator returns gti and stores (ti,att(y)) into ListH, where ti ∈R Fp.3) Key Queries. Here we combine the simulations of the algorithms CSP.KeyGen and KA.KeyGen as one key query. We note that it will not bring additional advantage for A in winning the game. When A queries a user i's secret key for an attribute set S, the simulator works as follows. It chooses ri,wj ∈R Fp, and computes D = gαgβri, L = gri, ∀ j ∈ S : Dj = H(j)riwj. The simulator finally sends the secret key to A and stores (SK,i,S) into ListSK. 4) Challenge. A outputs two equal length messages M0,M1 ∈ GT and an access tree A to the simulator, where there should not be any secret key issued to A such that the key satisfies A. The simulator chooses a s ∈R Fp, and next uses linear secret sharing technique to construct shares λy of s for all attributes y in A as in the algorithm DO.Encrypt, where λy is uniformly and independently random in Fp, and λy can be seen as a linear combination of independent random variables (in Fp) and s. The simulator then chooses θ ∈R Fp, and sets ˜ C = ˆ e(g,g)θ, C = gs, ∀ y ∈ Y, i ∈[1 ,n], Cy = gβλiH(att(y))−wis, and ∀j ∈ (1,n],C y,j = H(att(y))−(wj−wi)s, where wi,wj ∈R Fp. The simulator sends the challenge ciphertext to A.

5) Key Queries. Same as the previous key queries phase but with the restriction that the querying key cannot satisfy A. 6) Guess. A outputs a guess bit. Below we consider unexpected collision. An oracle query can be seen as a rational function ϑ = η/ψ in the variables θ, α, β, ti, wi, ri, λi and s. Suppose there are two distinct rational functions ϑ = η/ψ and ϑ′ = η′/ψ′. An unexpected collision event indicates that taking two different queries corresponding to the two functions, we have the same

output due to random choice of variables. If the event happens, it means that $\vartheta = \vartheta'$, and further $\eta\psi' - \eta'\psi = 0$. By the Schwartz-Zippel Lemma in [33], [40], the probability of the event is O(1/p). Therefore, the probability of a collision event is at most O(q2/p). Accordingly, the unexpected collision will not occur in the simulations with probability 1−O(q2/p). Remember that each group element is uniformly and de- pendently chosen in the above simulations. A can tell the difference between θ and αs elements in GT if there are two distinct queries ϑ and ϑ′ leading to the same output. Assume $\vartheta' = \gamma'\theta$ and $\vartheta = \gamma\alpha s$, we have $\vartheta - \vartheta' = \gamma\alpha s - \gamma'\theta$ such that $\gamma'\theta + \vartheta - \vartheta' = \gamma\alpha s$, where γ and γ′ are non-zero constant. We will show that A cannot construct a query for γαs in GT. We here observe all possible rational function queries in GT by means of bilinear map and the group elements given to A. It can be seen that A can obtain a transcript {g, gβ, gs, gβλig−tiwis, g−(wj−wi)sti, gαgβr, gr, grwiti} from onequery. For another transcript, we set it as {g, gβ, gs, gβλi′g−ti′wi′s, g−(wj′−wi′)sti′, gαgβr′, gr′, gr′wi′ti′}. We first ignore gβ since the elements with α will be tagged with β which is irrelevant to αs. To output a factor αs, we should focus on elements with factors α and s.It is not difficult to see that there are three types of outputs with αs in GT from two transcripts. One is αs+βrs (resp. αs+βr′s). To output γαs, we need an element βrs. However the element does not exist. The second format with αs is (−wj +wi)tiαs+(−wj +wi)stiβr′ (resp. (−wj′ + wi′)ti′αs + (−wj′ + wi′)sti′βr). To eliminate the part right after +, we need to concentrate on the elements with ti. The elements with βλi − tiwis, and rwiti fail to construct a cancel-out part as they are lack of a factor wj. For the element−(wj−wi)sti, we need an element with βr′ (resp. βr). But none of other elements satisfy our requirement. The last format with αs is −tiwiαs + β2λir′−tiwisβr′ + αβλi (resp. −ti′wi′αs+β2λi′r−ti′wi′sβr +αβλi′). If we cannot find elements to cancel out all of terms except for that of αs, it indicates thatAfails to construct γαs. For simplicity, we only check with the last term αβλi. It can be seen that α+βr is the only element with α. Thus, we need a βλi term. Nevertheless, the term does not exist. From the above observation, we can therefore state that A fails to construct the query form γαs. In addition, an improved key issuing protocol is proposed to resolve the key escrow problem of CP-ABE in cloud computing. In this paper, we assume that they do not collude with each other to share their master secret keys. We say that our proposed key issuing protocol is secure when the following two aspects are satisfied. The first one is that the KA cannot derive the user secret key if the CSP is honest. The other is that the CSP cannot derive the user secret key while the KA is honest. Security analysis about the protocol is described as below. Theorem 2. The proposed key issuing protocol in section IV-C is a secure protocol for computing gαhr by KA and CSP. Assume that the underlying arithmetic 2PC and zero knowledge proofs are secure, and (for security against corrupt CSP) that DDH is hard. Proof. First, to note that D = Y 1/ρ2 3 = X1/ρ2τ 2 = (Y ρ1 1 Y2)1/τ = Xρ1/β 1 hr = gα1+α2hr = gαgβr. To show the security we consider the cases of corrupting KA and corrupting CSP respectively. (1) For a corrupted KA, our simulator proceeds as follows: SimC : First, it will run the arithmetic 2PC simulator for computation of (α1 +α2)β. In the process, it will extract α1. Next, the simulator will choose random values X1 ∈G0, and senditto KA.It will receive Y1 and Y2 from thead versary KA, and two corresponding zero knowledge proofs. We will extract β and r from the corresponding proofs. Then, it will send α1, β and r to the trusted party, and receive gα2·gα1+βr = gα+βr, which will be CSP's secret key output. Consider a hybrid simulator HybC that takes as input of CSP's secret α2. It first runs the arithmetic 2PC simulator for the computation of x with the correct output value according to α2. Then the simulator completes the protocol as the honest CSP would do. This is clearly indistinguishable from the real CSP's protocol by the security of the arithmetic 2PC. Now, assuming that the proof of knowledge scheme is secure, HybC should be indistinguishable from the above simulator SimC. This is because the value X1 used by SimC will be distributed identically to those in HybC. (Since ρ1 is chosen at random in the real protocol, X1 will be distributed uniformly over G0 in the real protocol as in the simulated protocol.) Thus, interaction with our simulation is indistinguishable from interaction with an honest CSP. (2) For a corrupted CSP, our simulator proceeds as follows: SimK : First, it will run the arithmetic 2PC simulator for computation of (α1 + α2)β. This 2PC will extract α2 from CSP and output x = (α2 + α1)β mod p. We will choose a random value x ∈ Zp, and give it to the arithmetic 2PC simulator. Note that this is correctly distributed, since there is some β such that x = (α2 + α1)β mod p for any x,α1,α2. Next, our simulator will receive X1 from the adversary, and the corresponding zero knowledge proof. ρ1 is extracted by the proof system. We will select random values Y1,Y2 ∈G0, and send them to CSP. (Again, this will be distributed exactly as in a real execution.) We will receive X2 from the adversary, and use the corresponding proof to extract ρ2. Then, it will send α2 to the trusted party, and receive D = gα2 ·gα1+βr = gα+βr. Finally, it will compute Y3 = Dρ2 and send it to CSP. Consider a hybrid simulator HybK that takes as input of KA's secrets α1, β and r. It will compute x = (α2 + α1)β using the arithmetic 2PC simulator. When the 2PC simulator provides α2 and asks for output, it will correctly compute (α2 +α1)β. Then it will complete the execution as in the real protocol. This protocol is clearly indistinguishable from the real KA's protocol by security of the arithmetic 2PC.

In addition, we consider a second hybrid Hyb′K which is the same as the HybK to proceed the above protocol, but which uses the zero-knowledge simulator for all proofs of knowledge. This must be indistinguishable by the zero-knowledge property of the proof system. Here we only need show that the Hyb′K is indistinguishable from the interaction with the above simulator. Consider the reduction from DDH assumption: Given g, A = ga, B = gb, C = gc, and we must decide whether c = ab or c ∈R Zp. Here we define X1 = A = ga and h = gσ for σ ∈R Zp. As the SimK, we run the arithmetic 2PC simulator for computation of (α1 + α2)β and extraction α2. Next we receive X1 = A and extract ρ1 from the corresponding proof. Meanwhile, we compute Y1 = Xθ/β 1 = gaθ/β = C1/β, Y2 = hrθ = gσrθ = Bσr, and send them to the adversary CSP, along with a simulated proof of knowledge. Then we receive X2 and extract ρ2 from the proof. At last, we compute Y3 = X1/θ 2 = (Aρ1/β ·gθr)ρ2 and send it to CSP. Here we assume that the proofs of knowledge are secure. If c = ab, Y1, Y2, Y3 will be distributed correctly, and this will be distinguishable from Hyb′K. If c is a random number (that is, c ∈R Zp), then Y1,Y2 are randomly selected from G0, as in SimK. Thus, any adversary that can distinguish Hyb′K from SimK will allow us to resolve DDH problem. Under the DDH assumption, interaction with SimK is indistinguishable from interaction with a real KA.

VII. CONCLUSION

In this paper, we redesigned an attribute-based data shar- ing scheme in cloud computing. The improved key issuing protocol was presented to resolve the key escrow problem. It enhances data confidentiality and privacy in cloud system against the managers of KA and CSP as well as malicious system outsiders, where KA and CSP are semi-trusted. In addition, the weighted attribute was proposed to improve the expression of attribute, which can not only describe arbitrary- state attributes, but also reduce the complexity of access policy, so that the storage cost of ciphertext and time cost in encryp- tion can be saved. Finally, we presented the performance and security analyses for the proposed scheme, in which the results demonstrate high efficiency and security of our scheme.

ACKNOWLEDGMENTS This work was supported in part by the National Nat- ural Science Foundation of China (61171072, 61472083 and 61170283), in part by the Science and Technol- ogy Projects of Shenzhen (ZDSYS20140430164957660 and JCYJ20140418095735608), in part by the National High-Technology Research and Development Program ("863" Pro-gram) of China under Grant 2013AA01A212, and in part by the privacy-aware retrieval and modelling of genomic data (PRIGENDA, No. 13283250), the Academy of Finland.

REFERENCES

1. J. Baek, Q. H. Vu, J. K. Liu, X. Huang, and Y. Xiang. A secure cloud computing based framework for big data information management of smart grid. IEEE Transactions on Cloud Computing, 3(2):233–244, 2015.
2. A. Balu and K. Kuppusamy. An expressive and provably secure ciphertext-policy attribute-based encryption. Information Sciences, 276(4):354–362, 2014.
3. M. Belenkiy, J. Camenisch, M. Chase, M. Kohlweiss, A. Lysyanskaya, and H. Shacham. Randomizable proofs and delegatable anonymous credentials. Proceedings of the 29th Annual International Cryptology Conference, pages 108–125, 2009.
4. J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute- based encryption. IEEE Symposium on Security and Privacy, pages 321–334, 2007.
5. D. Boneh, B. Lynn, and H. Shacham. Short signatures from the weil pairing. Journal of Cryptology, 17(4):297–319, 2001.
6. M. Chase. Multi-authority attribute based encryption. Proceedings of the 4th Conference on Theory of Cryptography, pages 515–534, 2007.
7. M. Chase and S. S. Chow. Improving privacy and security in multi- authority attribute-based encryption. Proceedings of the 16th ACM Conference on Computer and Communications Security, pages 121–130, 2009.
8. L. Cheung and C. Newport. Provably secure ciphertext policy ABE. Proceedings of the 14th ACM conference on Computer and communi-cations security, pages 456–465, 2007.
9. S. S. Chow. Removing escrow from identity-based encryption. Pro- ceedings of the 12th International Conference on Practice and Theory in Public Key Cryptography, pages 256–276, 2009.
10. C. K. Chu, W. T. Zhu, J. Han, J. K. Liu, J. Xu, and J. Zhou. Security concerns in popular cloud storage services. IEEE Pervasive Computing, 12(4):50–57, 2013.
11. A. De Caro and V. Iovino. JPBC: java pairing based cryptography. IEEE Symposium on Computers and Communications, 22(3):850–855, 2011.
12. H. Deng, Q. Wu, B. Qin, J. Domingo-Ferrer, L. Zhang, J. Liu, and W. Shi. Ciphertext-policy hierarchical attribute-based encryption with short ciphertexts. Information Sciences, 275
13. Proceedings of the 13th ACM conference on Computer and communication (11):370–384, 2014.
14. C. Fan, S. Huang, and H. Rung. Arbitrary-state attribute-based en- cryption with dynamic membership. IEEE Transactions on Computers, 63(8):1951–1961, 2014.
15. V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryp- tion for fine-grained access control of encrypted data. Proceedings of the 13th ACM conference on Computer and communications security, pages 89–98, 2006.
16. J. Hur. Improving security and efficiency in attribute-based data sharing. IEEE Transactions on Knowledge and Data Engineering, 25(10):2271–2282, 2013.
17. L. Ibraimi, M. Petkovic, S. Nikova, P. Hartel, and W. Jonker. Mediated ciphertext-policy attribute-based encryption and its application. Pro- ceedings of the 10th International Workshop on Information Security Applications, pages 309–323, 2009.
18. T. Jiang, X. Chen, J. Li, D. S. Wong, J. Ma, and J. K. Liu. Towards secure and reliable cloud storage against data re-outsourcing. Future Generation Computer Systems, 52:86–94, 2015.
19. S. Lai, J. K. Liu, K.-K. R. Choo, and K. Liang. Secret picture: An efficient tool for mitigating deletion delay on OSN. Information and Communications Security, pages 467–477, 2015.
20. K. Liang, M. H. Au, J. K. Liu, W. Susilo, D. S. Wong, G. Yang, T. V. X. Phuong, and Q. Xie. A DFA-based functional proxy re-encryption scheme for secure public cloud data sharing. IEEE Transactions on Information Forensics and Security, 9(10):1667–1680, 2014.
21. K. Liang, M. H. Au, J. K. Liu, W. Susilo, D. S. Wong, G. Yang, Y. Yu, and A. Yang. A secure and expressive ciphertext-policy attribute-based proxy re-encryption for cloud data sharing. Future Generation Computer Systems, 52(C):95–108, 2015.
22. K. Liang, L. Fang, D. S. Wong, and W. Susilo. A ciphertext-policy attribute-based proxy re-encryption scheme for data sharing in public clouds. Concurrency and Computation: Practice and Experience, 27(8):2004–2027, 2015.
23. K. Liang, J. K. Liu, R. Lu, and D. S. Wong. Privacy concerns for photo sharing in online social networks. IEEE Internet Computing, 19(2):58– 63, 2015.
24. K. Liang, J. K. Liu, D. S. Wong, and W. Susilo. An efficient cloud- based revocable identity-based proxy re-encryption scheme for public clouds data sharing. Proceedings of the 19th European Symposium on Research in Computer Security, pages 257–272, 2014.
25. K. Liang and W. Susilo. Searchable attribute-based mechanism with efficient data sharing for secure cloud storage. IEEE Transactions on Information Forensics and Security, 10(9):1981–1992, 2015.
26. K. Liang, W. Susilo, and J. K. Liu. Privacy-preserving ciphertext multi- sharing control for big data storage. IEEE Transactions on Information Forensics and Security, 10(8):1578–1589, 2015