Zhu Y, Hu HX, Ahn GJ *et al.* Provably secure role-based encryption with revocation mechanism. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 26(4): 697–710 July 2011. DOI 10.1007/s11390-011-1169-9

Provably Secure Role-Based Encryption with Revocation Mechanism

Yan Zhu[1,2,*] (朱　岩), *Member*, *CCF*, Hong-Xin Hu[3] (胡宏新), Gail-Joon Ahn[3], *Senior Member*, *ACM*, *IEEE*
Huai-Xi Wang[4] (王怀习), and Shan-Biao Wang[4] (王善标)

[1] *Institute of Computer Science and Technology, Peking University, Beijing 100871, China*

[2] *Beijing Key Laboratory of Internet Security Technology, Peking University, Beijing 100871, China*

[3] *School of Computing, Informatics and Decision Systems Engineering, Arizona State University, Tempe, AZ 85287, U.S.A.*

[4] *School of Mathematical Sciences, Peking University, Beijing 100871, China*

E-mail:{yan.zhu, hxwang, xbwang}@pku.edu.cn; {hxhu, gahn}@asu.edu

Received December 5, 2010; revised May 15, 2011.

Abstract Role-Based Encryption (RBE) realizes access control mechanisms over encrypted data according to the widely adopted hierarchical RBAC model. In this paper, we present a practical RBE scheme with revocation mechanism based on partial-order key hierarchy with respect to the public key infrastructure, in which each user is assigned with a unique private-key to support user identification, and each role corresponds to a public group-key that is used to encrypt data. Based on this key hierarchy structure, our RBE scheme allows a sender to directly specify a role for encrypting data, which can be decrypted by all senior roles, as well as to revoke any subgroup of users and roles. We give a full proof of security of our scheme against hierarchical collusion attacks. In contrast to the existing solutions for encrypted file systems, our scheme not only supports dynamic joining and revoking users, but also has shorter ciphertexts and constant-size decryption keys.

Keywords cryptography, role-based encryption, role hierarchy, key hierarchy, collusion security, revocation

1 Introduction

Role-based access control (RBAC), as a proven alternative to traditional access controls, including discretionary access control (DAC) and mandatory access control (MAC), has been widely adopted in various information systems over the past few years[1]. Even though RBAC can tremendously help us minimize the complexity in administering users, it is still inevitable to adopt various cryptographic capabilities of managing resources in RBAC systems[2], so as to protect the resources that deviate from the access control system. However, the existing cryptographic schemes based on common asymmetric cryptosystem have several limitations to address above-mentioned features since those schemes cannot accommodate access control features of RBAC and lack scalability and interoperability for enterprise application environments with a large number of users[3].

Role-Based Cryptosystem, proposed by Zhu *et al.*[4], is a key management system that realizes encryption, signature and authentication according to role hierarchy (RH) in RBAC model. As an important part of role-based cryptosystem, Role-Based Encryption (RBE) enables an access control mechanism over encrypted data by hiding access permissions and assigned roles into private keys and ciphertexts. One of the advantages of RBE is the ability to easily link up with the existing RBAC models and systems. In an RBE system, a resource owner specifies an access permission, which could be a security clearance (SC) requirement, to encrypt the resource directly by encryption algorithm (which can be run by anyone knowing the universal public key issued by an authority). Each user in this system possesses a unique private key associated with a role, stating the identity and privilege level within the organization. Such a user can decrypt a ciphertext if the user's privilege level is equal to or higher than the clearance requirement associated with the ciphertext. Different from traditional cryptosystems, a key feature of RBE is that the encryption algorithm is not for individual user, but for the group of authorized users.

Regular Paper

This work of Yan Zhu, Huai-Xi Wang and Shan-Biao Wang were partially supported by the National Development and Reform Commission under Project "A Cloud-based service for monitoring security threats in mobile Internet" and "A monitoring platform for web safe browsing". This work of Gail-J. Ahn and Hong-Xin Hu were partially supported by the National Science Foundation of USA under Grant Nos. NSF-IIS-0900970 and NSFCNS-0831360.

*Corresponding Author

Role hierarchy is a natural means of structuring roles to reflect an organization's lines of authority and responsibility. In RBAC model, role hierarchy defines an inheritance relationship among roles, which is required to be a partial order. As role hierarchy is the foundation of RBAC, it is necessary to implement such a mechanism in the construction of RBE. It requires us not only to hide the information of RH into public encryption keys and user's keys, but also to find an efficient cryptographical method to validate the partial ordering relation among these keys. Moreover, it is necessary for practical applications to ensure shorter length of user's key, as well as to support a large number of users.

1.1 Related Work

The research on cryptographic hierarchical structures has a long history since hierarchical structure is a natural way to organize and manage a large number of users. Several approaches on cryptographic partial order relation supporting hierarchical structures have been proposed. Akl and Taylor introduced a simple scheme to solve multilevel security problem[5-6]. Since then, several efficient methods have been studied. The concept of logical key hierarchy (LKH) was proposed by Wallner *et al.*[7] and Wong *et al.*[8] independently. In this paradigm, common encryption key was organized into a tree structure to achieve secure group communication in a multicast environment. Several modifications have been proposed, such as complete subtree (CS)[9], subset difference (SD)[10], and layered subset difference (LSD)[11]. All of these schemes are constructed in symmetric-key setting, and the lengths of ciphertext and user's key are directly associated with the number of users.

Public-key hierarchical cryptosystems have been recently proposed, e.g., Boneh and Franklin proposed the first fully functional identity-based encryption scheme (IBE)[12-13] in 2001, in which the public key can be an arbitrary string such as an email address. In order to manage a large number of users, hierarchical identity-based encryption (HIBE) mirrors an organizational hierarchy[14]. Another important area is hierarchy key management (HKM) that also organizes the key into a hierarchy. For example, time-bound hierarchical key assignment (THKA)[15] can assign time-dependent encryption keys to a set of classes in a partially ordered hierarchy. This scheme is especially suitable for the realtime broadcast system with time control. Since all users with the same role share the same key, these schemes are hard to realize certain useful security mechanisms, such as revocation, digital forensics, and traceability.

Since Sahai and Waters[16] introduced attribute-based encryption (ABE) as a new means of encrypted access control in 2005, ABE has received much attention and many schemes have been proposed recently, such as, key-policy ABE (KP-ABE)[17-19], ciphertext-policy ABE (CP-ABE)[20-23] and dual policy ABE (DP-ABE)[24-25]. ABE is based on the attribute-based access control (ABAC)[26-27] or fine grained access control (FGAC)[28-29] model. In an ABE system, ciphertexts are not necessarily encrypted to one particular user by an assigned key as in traditional public key cryptography. Instead, both users' private keys and ciphertexts will be associated with a set of attributes or a policy over attributes. A user is able to decrypt a ciphertext if there is a "match" between his private key and the ciphertext. Although it takes time to widely adopt ABAC or FGAC model in practical management information systems (MIS), it is worth learning the features of access policy representation in ABE. At the same time, the research on attribute-based signature has also made great progress and various attribute-based signature schemes[30-32] have been proposed. There are several recent studies of revocation mechanism of ABE. For example, Attrapadung and Imai[33] presented a revocation approach for a subset of at most t users based on Linear Secret Sharing Scheme (LSSS) with random polynomial of degree t. This approach also leads to a longer size of public-key, which is direct proportional to the number of revoked users t. Thus, it is not an effective solution to practical applications, especially to supporting a large number of users.

To overcome the limitations of the exiting solutions, we presented a practical cryptographic RBAC model in [4], called role-key hierarchy model, to support various security features based on role-key hierarchy. In that work, we introduced a role-based cryptosystem construction, which includes a role-based encryption (RBE) scheme and a role-based signature (RBS) scheme. With the help of role-key hierarchy, this RBE scheme can be tightly integrated into the hierarchical RBAC systems. However, this RBE scheme could not support the revocations of users and roles. Thus, efficient revocation mechanism should be introduced to improve the existing RBE scheme.

1.2 Contributions

In this paper, we focus on the construction of a cryptosystem compatible with hierarchical RBAC model. With the help of bilinear pairings, we present an enhanced *Role-Based Encryption* with revocation mechanism. Our new scheme provides more flexible control than other schemes, as well as an efficient revocation mechanism to support any number of users (or identities) and roles. Furthermore, our scheme has following

Table 1. Comparison Between Role-Based Encryption and Attribute-Based Encryption

	RBE (our work)	CP-ABE (Bethencourt *et al.*'s work)[20]
Access Model	RBAC	ABAC[26-27] and FGAC[28-29]
Access Policy	Partial order relation of role hierarchy based on RBAC	Policy based on the set of attributes, which does not involve partial order relation
Policy Structure	Lattice (tree and inverse tree)	Tree
Public Key	System parameters, role hierarchy and user label set	System parameters and attribute set
Private Key	Constant size, specified by role	Variable length, specified by a set of attributes
Ciphertext	Variable length, proportional to authorized role and revoked user set	Variable length, proportional to the number of leaf nodes in the policy tree
Encryption	Policy is implemented by RBAC system	Policy is specified by the encryptor
Decryption	Matching between derived role and assigned role in ciphertext and aggregate algorithm	Policy tree retracing and matching between the user's attribute and assigned attribute in ciphertext
Revocation	Dynamic user revocation	None
Main Techniques	Partial ordering and bilinear map	Secret sharing and bilinear map

new properties: key hierarchy can support arbitrary partial-order structures and an unlimited number of roles; a manager can dynamically add infinitely many users without revising the existing ciphertexts and user's private keys; and encryption is collusion-secure for arbitrarily large collusion of users. Moreover, our construction also achieves an optimal bound of overhead rate for both ciphertexts and decryption keys. Most importantly, our RBE scheme has better performance and scalability than existing solutions for encrypted file systems (EFS).

To explain the features of RBE, in Table 1 we show the main differences between RBE and ABE in comparison with BSW's CP-ABE[20]. For example, our RBE scheme has a user's private key with constant size rather than variable size; the RBE encryption is automatically performed by RBAC systems, which serve to reduce the burdens of regulation on managements; there exists an efficient revocation method for a subset of users; and so on.

The rest of the paper is organized as follows. Section 2 overviews some basic notions and complexity assumptions. Section 3 articulates the definition of RBE and security models. In Section 4, we address our RBE construction. We evaluate the security and performance of our schemes in Sections 5 and Section 6, respectively. Finally, we conclude this paper in Section 7.

2 Preliminaries

In this section, we present a brief description of partial order relation and role hierarchy in RBAC model. Next, we briefly review the necessary facts about bilinear maps, as well as a class of assumptions used in our RBE scheme.

2.1 Partial Order Relation and Role Hierarchy

Let $\Psi = \langle P, \preceq \rangle$ be a (finite) partially ordered set (Poset) with partial order relation $\preceq$ on a (finite) set P. A partial order is a reflexive, transitive and anti-symmetric binary relation. We provide some terminology for partial order relation. Two distinct elements x and y in Ψ are said to be comparable if $x \preceq y$ or $y \preceq x$. Otherwise, they are incomparable, denoted by $x \| y$. An order relation $\preceq$ on P gives rise to a relation $\prec$ of strict partial order: $x \prec y$ in P iff $x \preceq y$ and $x \neq y$. We define the predecessors and successors of elements in $\Psi = \langle P, \preceq \rangle$ as follows: for an element x in P, $\uparrow x = \{y \in P : x \preceq y\}$ denotes the set of predecessors of x, $\downarrow x = \{y \in P : y \preceq x\}$ denotes the set of successors. Two posets are said to be isomorphic if their "structures" are entirely identical. Formally, posets $\Psi = \langle P, \preceq \rangle$ and $\Phi = \langle Q, \preceq \rangle$ are isomorphic if there exists a bijection f from P to Q such that $x \preceq y$ iff $f(x) \preceq f(y)$.

In hierarchical RBAC model, inheritance is reflexive because a role inherits its own permissions, and transitivity is a natural requirement in this context. Also, anti-symmetry rules out roles that inherit from one another and would therefore be redundant. For example, we can represent $\Psi = \langle P, \preceq \rangle$ by using circles (indicating the elements of P) and inter-connecting lines (indicating the covering relations). For example, Fig.1(a) shows the diagrams for some simple ordered sets. Sub-figures (a1), (a2) and (a3) are linear, tree, and inverted-tree structure, respectively. Sub-figure (a4) is a three-layer tree structure, in which the senior r_a (top) inherits the permissions from all other juniors (bottom). Especially, there exist two different paths $\{r_f, r_b, r_a\}$ and $\{r_f, r_c, r_a\}$ from r_f to r_a. Sub-figure (a5) is a three-layer hybrid structure, which is composed of various

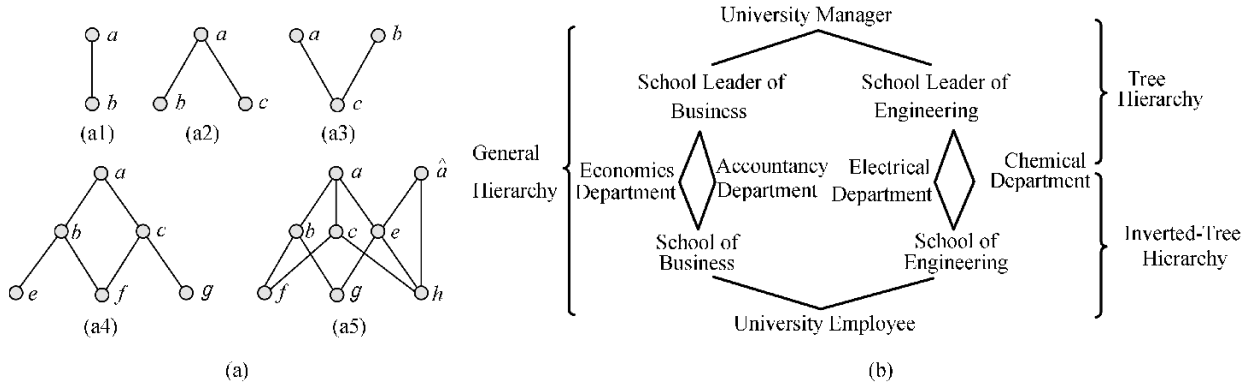

Fig.1. Diagrams for partial-order sets (a) and instance of role hierarchy (b).

different structures. Note that r_a and $r_{\hat{a}}$ are not related through hierarchy.

Generally, a hierarchy in RBAC is mathematically a partial order that defines an inheritance (or seniority) relation between roles, whereby senior roles acquire the permissions from their juniors. An example of role hierarchy is shown in Fig.1(b), in which more powerful (senior) roles are shown toward the top of the diagram and less powerful (junior) roles toward the bottom. Based on this, we present the definition of role hierarchy as follows.

Definition 1 (Role Hierarchy). *Given a set of users U, a role hierarchy $\mathcal{H}$ is a triple $\langle U, R, \preceq \rangle$, if there exists a (finite) partially ordered set $\langle R, \preceq \rangle$, such that each user belongs to and only belongs to a role, i.e., for all $u_{i,j} \in U$, there exists an $r_i \in R$, such that $u_{i,j} \in r_i$.*

2.2 Bilinear Maps and Some Assumptions

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two additive groups and $\mathbb{G}_T$ be a multiplicative group with large prime order p.[①] A computable bilinear map is a function $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ with the following properties: for any $G \in \mathbb{G}_1$, $H \in \mathbb{G}_2$ and all $a, b \in \mathbb{Z}_p$, we have 1) bilinearity: $e([a]G, [b]H) = e(G, H)^{ab}$; 2) non-degeneracy: $e(G, H) \neq 1$ unless G or $H = 1$; 3) computability: $e(G, H)$ is efficiently computable. A bilinear map group system is a tuple $\mathbb{S} = \langle p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e \rangle$ composed of the objects described above.

Security of our system is based on a complexity assumption called the General Decisional bilinear Diffie-Hellman Exponent (GDDHE) assumption. We define the GDDHE problem as follows.

Definition 2 (GDDHE Problem). *Let $F_1, F_2, F_3 \in \mathbb{Z}_p[X_1, \ldots, X_m]^s$ be three s-tuples of m-variate polynomials over $\mathbb{Z}_p$, where $s, m \in \mathbb{Z}^+$. Given a vector*

$$\boldsymbol{H}(x_1, \ldots, x_m) = \begin{pmatrix} [F_1(x_1, \ldots, x_m)]G, \\ [F_2(x_1, \ldots, x_m)]H, \\ e(G, H)^{F_3(x_1, \ldots, x_m)} \end{pmatrix} \in \mathbb{G}_1^s \times \mathbb{G}_2^s \times \mathbb{G}_T^s,$$

and $T \in \mathbb{G}_T$, decide whether $T = e(G, H)^{h(x_1, \ldots, x_m)}$, where $h \in \mathbb{Z}_p[X_1, \ldots, X_m]$.

We refer to Theorem A.2 as a proof that GDDHE has generic security when $h \notin (F_1, F_2, F_3)$ in [34]. In Lemma 1, we restate the generic security in a more concrete form for (n, t)-GDDHE$_1$ problem[34-35].

Definition 3 ((n, t)-GDDHE$_1$ Problem). *Let $f(x)$ and $g(x)$ be two known random polynomials of degree t and $n - t$ with pairwise distinct roots respectively,*

$$\begin{cases} f(x) = \prod_{i=1}^{t} (\zeta_i x + x_i) = \sum_{i=0}^{t} a_i \cdot x^i \\ g(x) = \prod_{i=1}^{n-t} (\zeta_{t+i} x + x'_i) = \sum_{i=0}^{n-t} b_i \cdot x^i \end{cases} \quad mod\ p$$

where $\prod_{i=1}^{t} \zeta_i = 1$ and $\prod_{i=1}^{n-t} \zeta_{t+i} = 1$ mod p. Let $h(x, y) = yf(x)g(x)$ be a two-variable polynomial, and $\mathbb{S} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e(\cdot, \cdot))$ be a group system. Given the values in (F_1, F_2, F_3, T)-GDDHE problem with

$$\begin{cases} F_1(\gamma, \varsigma) = \begin{pmatrix} G, [\gamma]G, \cdots, [\gamma^{t-1}]G, \\ [\gamma \cdot f(\gamma)]G, [\varsigma \cdot \gamma \cdot f(\gamma)]G \end{pmatrix}, \\ F_2(\gamma, \varsigma) = \begin{pmatrix} H, [\gamma]H, \cdots, [\gamma^n]H, \\ [\varsigma \cdot g(\gamma)]H \end{pmatrix}, \\ F_3(\gamma, \varsigma) = e(G, H)^{f^2(\gamma) g(\gamma)}, \end{cases} \tag{1}$$

and $T \in \mathbb{G}_T$, decide whether $e(G, H)^{\varsigma \cdot f(\gamma) \cdot g(\gamma)} = T$, where $\gamma, \varsigma, \zeta_i, x_i, x'_i \in \mathbb{Z}_p^*$ are two secret random variables and G, H are generators of $\mathbb{G}_1, \mathbb{G}_2$, respectively.

[①] We require that no efficient isomorphism $\mathbb{G}_2 \to \mathbb{G}_1$ or $\mathbb{G}_1 \to \mathbb{G}_2$ is known, or $\mathbb{G}_2 \to \mathbb{G}_1$ is known but its inverse $\mathbb{G}_1 \to \mathbb{G}_2$ is unknown.

3 Definitions

In this section, we begin by formally defining what is a role-based encryption system. We then state the security requirements and adversary's attack models needed for our proof of security. For the sake of clarity, we list some notations used throughout this paper in Table 2.

Table 2. Description of the Notations

No.	Notation	Description
1	κ	Security parameter
2	U, R	U and R denote the set of users and roles, respectively
3	$\mathcal{H}, \mathcal{R}$	$\mathcal{H}$ denotes the hierarchy $\langle U, R, \preceq \rangle$ and $\mathcal{R}$ denotes the set of revoked users
4	par, mk	par and mk denote the public parameter and the master key, respectively
5	$r_i, u_{i,j}$	r_i denotes the i-th role in R and $u_{i,j}$ denotes the j-th user with role r_i
6	pk_i	Public role key for $r_i \in R$
7	$dk_{i,j}$	Decryption key of user $u_{i,j}$
8	$M, \mathcal{C}_i$	M and $\mathcal{C}_i$ denote the plaintext and the ciphertext encrypted by pk_i, respectively

3.1 Public-Key Role-Based Encryption

Given a role hierarchy $\mathcal{H} = \langle U, R, \preceq \rangle$, a public-key Role-Based Encryption (RBE) is specified by five polynomial-time algorithms $\langle S, G, A, E, D \rangle$.

1) *Setup*$(\kappa, \mathcal{H}) \rightarrow (mk, par)$. Take a security parameter κ and a role hierarchy $\mathcal{H}$ as input. It produces a manager key mk and a public parameter par.

2) *GenRKey*$(par, r_i) \rightarrow pk_i$. Take the parameter par and a role index r_i. It generates a public encryption key pk_i of r_i.

3) *AddUser*$(mk, ID, r_i, u_{i,j}) \rightarrow (lab_{i,j}, dk_{i,j})$. Take a user identity ID, a user index $u_{i,j}$ in the role r_i, and the manager key mk. It outputs a user's secret key, which involves a private key $dk_{i,j}$, a user label $lab_{i,j}$, and $par = par \bigcup \{lab_{i,j}\}$.

4) *Encrypt*$(\mathcal{R}, pk_i, M) \rightarrow \mathcal{C}_i$. Take as input the encryption key pk_i, a message M and a set of revoked users $\mathcal{R} \subseteq U$. It returns a ciphertext $\mathcal{C}_i$.

5) *Decrypt*$(\mathcal{R}, dk_{i,j}, \mathcal{C}_i) \rightarrow M$. Take as input a ciphertext $\mathcal{C}_i$, a subset $\mathcal{R} \subseteq \mathcal{H}$ and a user decryption key $dk_{i,j}$. It returns a message M.

We need to realize the user revocation by the user label $lab_{i,j}$, called identity-based revocation (IBR). With the help of this revocation mechanism, some users $\{u_{i,j}\} \in \mathcal{R}$ can be revoked temporarily from the authorized users in ciphertexts. This paper does not highlight the revocation of roles since we can realize this mechanism by using *Control Domain* addressed in Section 4.

Given an instance of RBE scheme $\mathcal{E}$ under a certain $\mathcal{H}$, we define a key hierarchy $\mathcal{K} = \{UK, RK, \preceq\}$ from $(\mathcal{E}, \mathcal{H})$, where $UK = \{dk_{i,j}\}_{\forall dk_{i,j} \leftarrow A(\cdot)}$, $RK = \{pk_i\}_{\forall pk_i \leftarrow G(\cdot)}$, and $pk_i \preceq pk_j$ iff there exists a polynomial-time algorithm $F(\mathcal{H}, pk_i, r_j) = pk_j$. We call F the derivation (or delegation[36]) function of $\mathcal{E}$, which is used to realize the partial order relation in a set of public encryption keys.

Note that, we require that a user belongs to a single role rather than to multiple roles in this definition due to the construction limitations in cryptography. Although this requirement is not true in general RBAC model, it is necessary to require strict role-based authentication mechanisms to provide strong security. If necessary the manager can assign multiple secret keys to different roles, but they have the same label. In addition, in practice the RBAC model can automatically employ the user's current role to invoke the *Encrypt* algorithm.

3.2 Security Notions and Adversary's Attack Models

The security requirements of RBE system, made up of three properties, are defined as follows.

Definition 4 (RBE). *Given a role hierarchy* $\mathcal{H} = \langle U, R, \preceq \rangle$, *an RBE scheme* $\mathcal{E} = \langle S, G, A, E, D \rangle$ $(|R| = m, |U| = n)$ *satisfies the following conditions.*

1) *Consistency*: *The representations are equivalent between the role hierarchy* $\mathcal{H}$ *and the reduced key hierarchy* $\mathcal{K}$, *that is,* $\{pk_i \preceq pk_j\}_{\mathcal{K}} \sim \{r_i \preceq r_j\}_{\mathcal{H}}$, *where* $\sim$ *denotes isomorphism.*

2) *Viability*: *For every set of revoked users* $\mathcal{R}$, $M \in \mathcal{M}$, *and* $\mathcal{C}_i = E(\mathcal{R}, pk_i, M)$,

$$\Pr\left[\begin{array}{c} D(\mathcal{R}, dk_{j,l}, \mathcal{C}_i) = M : \\ \forall u_{j,l} \in r_j, r_i \preceq r_j \wedge u_{j,l} \notin \mathcal{R} \end{array}\right] = 1.$$

3) *Security*: *For any probabilistic polynomial-time algorithm* D', *every polynomial* $p(\cdot)$, *all sufficiently large* $k \in \mathbb{N}$, *every* $\mathcal{R}$, $M \in \mathcal{M}$, *and* $\mathcal{C}_i = E(\mathcal{R}, pk_i, M)$,

$$\Pr\left[\begin{array}{c} D'(\mathcal{R}, dk_{j,l}, \mathcal{C}_i) = M : \\ \forall u_{j,l} \in r_j, r_i \not\preceq r_j \vee u_{j,l} \in \mathcal{R} \end{array}\right] < \frac{1}{p(k)}.$$

The principal attack on RBE system is the collusion attack between different users, that is, $\mathcal{A}$ corrupts some $u_{j,l} \in \mathcal{R}$ to decrypt $\mathcal{C}_i$, even if $u_{j,l} \in r_j$ and $r_i \preceq r_j$. Hence, we define a semantic security under Chosen Plaintext Attack against Hierarchical Collusion (denoted by IND-hcCPA). Security is defined using the following game between an attack algorithm $\mathcal{A}$ and a challenger $\mathcal{B}$. This game is defined as follows.

1) *Initial*. $\mathcal{B}$ constructs an arbitrary $\mathcal{H}$ $(|R| = m)$,

and then runs Setup algorithm and gives $\mathcal{A}$ the resulting parameters *par* and $\mathcal{H}$, keeping *mk* secret.

2) *Learning.* $\mathcal{A}$ adaptively issues n queries $q_1, \ldots, q_n$ to add the users and gets a set of collusion users $\mathcal{R}$ ($|\mathcal{R}| = t$) as follows.

(a) *Public Label Query* ($u_{i,j} \notin \mathcal{R}$): following $AddUser(mk, u_{i,j})$, $\mathcal{B}$ generates a user label $lab_{i,j}$ and sends it to $\mathcal{A}$;

(b) *Private Key Query* ($u_{i,j} \in \mathcal{R}$): following $AddUser(mk, u_{i,j})$, $\mathcal{B}$ generates a revoked user and returns this user's $lab_{i,j}$ and $dk_{i,j}$ to $\mathcal{A}$.

3) *Challenge.* $\mathcal{A}$ chooses two equal length plaintexts $M_0, M_1 \in \mathcal{M}$ and appoints a role r_i on which it wishes to be challenged. $\mathcal{B}$ picks a random bit $b \in \{0, 1\}$ and sends the challenge ciphertext $E(\mathcal{R}, pk_i, M_b)$ to $\mathcal{A}$.

4) *Guess.* $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$ for b, and wins if $b = b'$.

The above game models an attack where all users, who are not in the set $\mathcal{R}$, collude to try and expose a ciphertext intended for users in $U \setminus \mathcal{R}$ only. The set $\mathcal{R}$ is chosen by the adversary. In this game, we define the advantage of the adversary $\mathcal{A}$ in attacking the scheme as

$$\begin{aligned} Adv^{\mathrm{ind}}_{\mathcal{E},\mathcal{A}}(m, n, t) &= |\Pr[b' = b] - \Pr[b' \neq b]| \\ &= |2\Pr[b' = b] - 1|, \end{aligned}$$

where $|\mathcal{R}| = t$, $|R| = m$, $|U| = n$, and the probability is taken over the random coins of $\mathcal{A}$ and all probabilistic algorithms in the scheme.

Definition 5 (Secure Role-Based Encryption). *An RBE scheme $\mathcal{E}$ is said to be an (m, n, t)-secure role-based encryption if for any polynomial-time adversary $\mathcal{A}$, the total number of roles m, the total number of users n, and at most t colluders, any computational advantage of adversary $Adv^{\mathrm{ind}}_{\mathcal{E},\mathcal{A}}(m, n, t)$ is negligible in the above IND-hcCPA game. The scheme $\mathcal{E}$ is said to be semantically secure against full collusion if it is (m, n, n)-secure.*

4 Our Construction

In this section, we describe a public-key RBE scheme with role hierarchy, which has new features including $O(m)$-size ciphertexts and encryption key, as well as $O(1)$-size decryption key for the number of roles m. This construction also supports the revocation of any number of users.

4.1 Role-Based Encryption Scheme

Let $\mathcal{H} = \{U, R, \preceq\}$ be a role-key hierarchy. Without loss of generality, we assume that the total number of roles is m in $\mathcal{H}$, i.e., $R = \{r_1, r_2, \ldots, r_m\}$. We construct an RBE scheme as follows.

1) *Setup*(κ, Ψ). Let $\mathbb{S} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ be a bilinear map group system with randomly selected generators $G \in \mathbb{G}_1$ and $H \in \mathbb{G}_2$ respectively, where $\mathbb{G}_1$, $\mathbb{G}_2$ are two bilinear groups of prime order p, $|p| = O(\kappa)$. This algorithm first picks a random integer $\tau_i \in \mathbb{Z}_p^*$ for each r_i in role-key hierarchy graph. We define

$$\begin{aligned} &D_i = [\tau_i]G \in \mathbb{G}_1, \quad \forall r_i \in R, \\ &V = e(G, H) \in \mathbb{G}_T, \end{aligned}$$

where τ_i is the secret of each role r_i and D_i is called the identity of this role. Furthermore, it defines $D_0 = [\tau_0]G$ by using a random $\tau_0 \in \mathbb{Z}_p^*$. Thus, the public parameter is $par = \langle H, V, D_0, D_1, \ldots, D_m \rangle$ and we keep $mk = \langle G, \tau_0, \tau_1, \ldots, \tau_m \rangle$ secret.

2) *GenRKey*(par, r_i). This is an assignment algorithm for role key from the public parameter *par*. For a role r_i, the role key pk_i can be computed as

$$\begin{aligned} &pk_i = \langle H, V, W_i, \{D_k\}_{\forall r_k \in \uparrow r_i} \rangle, \\ &W_i = D_0 + \sum_{r_i \not\preceq r_k} D_k \in \mathbb{G}_1, \end{aligned}$$

where $\{D_k\}_{\forall r_k \in \uparrow r_i}$ is the identity set of all roles in $\uparrow r_i$. It is clear that $W_i = [\tau_0 + \sum_{r_i \not\preceq r_k} \tau_k]G$. For sake of simplicity, let $\zeta_i = \tau_0 + \sum_{r_i \not\preceq r_k} \tau_k$, so we have $W_i = [\zeta_i]G$.

3) *AddUser*($mk, ID, r_i, u_{i,j}$). Given the manager key $mk = \langle G, \{\tau_i\}_{i=0}^m \rangle$ and a user index $u_{i,j}$ in the role r_i, the manager generates a unique decryption key by randomly selecting a fresh $x_{i,j} = Hash(ID, u_{i,j}) \in \mathbb{Z}_p^*$ and defining a public user label $lab_{i,j} = \langle x_{i,j}, V_{i,j}, B_{i,j} \rangle$ and a decryption key $dk_{i,j} = A_{i,j}$, where

$$\begin{aligned} x'_{i,j} &= x_{i,j} - \sum_{r_i \not\preceq r_k} \tau_k \in \mathbb{Z}_p^*, \\ A_{i,j} &= \Big[\frac{x'_{i,j}}{\zeta_i + x'_{i,j}}\Big] G \in \mathbb{G}_1, \\ B_{i,j} &= \Big[\frac{1}{\zeta_i + x'_{i,j}}\Big] H \in \mathbb{G}_2, \\ V_{i,j} &= V^{\frac{1}{\zeta_i + x'_{i,j}}} \in \mathbb{G}_T. \end{aligned}$$

Note that, the total number of users is unlimited in each role.

4) *Encrypt*($\mathcal{R}, pk_i, M$). To encrypt the message $M \in \mathbb{G}_T$, given any $pk_i = \langle H, V, W_i, \{D_k\}_{r_k \in \uparrow r_i} \rangle$ and a set of revoked users $\mathcal{R} = \{u_{i_1,j_1}, \ldots, u_{i_t,j_t}\}$, the algorithm randomly picks $\xi \in \mathbb{Z}_p^*$ and then computes

$$\begin{cases} C_1 = [\xi]W_i \in \mathbb{G}_1, \\ C_2 = [\xi]B_{\mathcal{R}} \in \mathbb{G}_2, \\ C_3 = M \cdot (V_{\mathcal{R}})^{\xi} \in \mathbb{G}_T, \\ C_4 = \{[\xi]D_k\}_{\forall r_k \in \uparrow r_i} \in \mathbb{G}_1^{\bar{m}}, \end{cases} \tag{2}$$

where, $\bar{m}$ is the number of elements in C_4, $|\mathcal{R}| = t$, and

$$B_{\mathcal{R}} = \begin{cases} H, & \text{if} \quad \mathcal{R} = \varnothing, \\ \Big[\dfrac{1}{\prod_{l=1}^{t}(\zeta_{i_l} + x'_{i_l,j_l})}\Big]H, & \text{if} \quad \mathcal{R} \neq \varnothing, \end{cases}$$

$$V_{\mathcal{R}} = \begin{cases} V, & \text{if} \quad \mathcal{R} = \varnothing, \\ V^{\frac{1}{\prod_{l=1}^{t}(\zeta_{i_l} + x'_{i_l,j_l})}}, & \text{if} \quad \mathcal{R} \neq \varnothing. \end{cases}$$

$B_{\mathcal{R}}$ and $V_{\mathcal{R}}$ can be efficiently computed by the aggregate algorithms from $\{B_{i_l,j_l}\}_{u_{i_l,j_l}\in\mathcal{R}}$ and $\{V_{i_l,j_l}\}_{u_{i_l,j_l}\in\mathcal{R}}$ (see Subsection 5.2.1). Finally, it outputs the ciphertext $\mathcal{C}_i = \langle C_1, C_2, C_3, C_4, \mathcal{R}\rangle$.

5) *Decrypt*$(dk_{j,k}, \mathcal{C}_i)$. Given a ciphertext $\mathcal{C}_i$ from the role r_i, the user $u_{j,k} \in r_j$ can utilize the following equation to recover M from $\mathcal{C}_i$ with private key $dk_{j,k} = A_{j,k}$ when $r_i \preceq r_j$ and $u_{j,k} \notin \mathcal{R}$:

$$V' = e\Big(C_1 + \sum_{r_l \in \Gamma(r_j, r_i)} D'_l, B^{\mathcal{R}}_{j,k}\Big) \cdot e(A_{j,k}, C_2), \quad (3)$$

where $\Gamma(r_j, r_i)$ denotes $\cup_{r_i \preceq r_l, r_j \not\preceq r_l}\{r_l\}$, $D'_l = [\xi]D_l \in C_4$ for all $r_l \in \Gamma(r_j, r_i)$, and

$$B^{\mathcal{R}}_{j,k} = \begin{cases} B_{j,k}, & \text{if} \quad \mathcal{R} = \varnothing, \\ \Big[\dfrac{1}{\prod_{l=1}^{t}(\zeta_{i_l} + x'_{i_l,j_l}) \cdot (\zeta_j + x'_{j,k})}\Big]H, & \text{if} \quad \mathcal{R} \neq \varnothing, \end{cases}$$

from $\{B_{i_l,j_l}\}_{u_{i_l,j_l}\in\mathcal{R}}$ and $B_{j,k}$. Finally, it outputs the plaintext $M = C_3/V'$.

The derivation function of public keys, $F(pk_i, r_l) = pk_l$ for any $r_i \preceq r_l$, can be defined as $F(pk_i, r_l) = \langle H, V, W_l, \{D_k\}_{\forall r_k \in \uparrow r_l}\rangle = pk_l$, where $W_l = W_i + \sum_{r_k \in \Gamma(r_l, r_i)} D_k$, due to $r_k \in \Gamma(r_l, r_i) \subseteq (\uparrow r_i)$ when $r_i \preceq r_l$. Note that, the part C_4 of $\mathcal{C}_i$ is called *Control Domain* of this ciphertext in (2). We can deal with access control constraints for roles by choosing the appropriate $r_k \in \uparrow r_i$ to insert into C_4. Furthermore, access control constraints for users can be effectively carried out by using the set of revoked users $\mathcal{R}$. The number of revoked users is unlimited in this scheme. Hence, we can revoke any subgroup of roles and users in terms of these two mechanisms.

5 Security Analysis

5.1 Analysis of Consistency

Since D_i is chosen at random, we need to consider the collision probability among the role keys $\{pk_i\}_{r_i \in R}$, i.e., $W_i = W_j$ for $i \neq j$, $W_i \in pk_i$, and $W_j \in pk_j$. The following theorem tells us that this collision probability is negligible if the security parameter κ is large enough.

Theorem 1. *The collision probability among m integers chosen from $\mathbb{Z}_p^*$ at random is less than $\frac{(m+1)^2}{4p}$.*

Proof. Firstly, the collision probability between λ random integers $\{a_i\}_{i=1}^{\lambda}$ and μ random integers $\{b_i\}_{i=1}^{\mu}$, $\sum_{i=1}^{\lambda} a_i = \sum_{j=1}^{\mu} b_j$, is $\frac{1}{p}$, where $a_1, \ldots, a_\lambda \in \mathbb{Z}_p^*$ and $b_1, \ldots, b_\mu \in \mathbb{Z}_p^*$. Secondly, the number of all possible unordered pairs $\{\lambda, \mu\}$ with $\lambda + \mu = k$ is $\lfloor \frac{k}{2} \rfloor$ for $1 \leqslant \lambda, \mu < m$. Thus the number of all possible unordered pairs $\{\lambda, \mu\}$ with $3 \leqslant \lambda + \mu \leqslant m$ is $\sum_{k=3}^{m} \lfloor \frac{k}{2} \rfloor < \sum_{k=1}^{m} \frac{k}{2} = \frac{m(m+1)}{4} < \frac{(m+1)^2}{4}$. Hence, with the help of Bernoulli's inequality, the collision probability is $1 - (1 - \frac{1}{p})^{\frac{(m+1)^2}{4}} \leqslant \frac{(m+1)^2}{4}\frac{1}{p} = \frac{(m+1)^2}{4p}$. Note that we do not assume that a_i and b_j are different. □

Since the total number of roles is far less than the size of space of keys, this theorem means that the collision probability is negligible for $m \ll p$, e.g., given $m = 1000$ and $|p| = 2 \times \kappa = 160$ (κ = 80-bits), the collision probability is less than $\frac{2^{20}}{2^{162}} = 2^{-142}$. This implies that different roles almost always have different keys. So we will neglect the collision probability hereinafter.

Theorem 2. *Under the above assignment, $\cup_{r_j \not\preceq r_k}\{\tau_k\} \subset \cup_{r_i \not\preceq r_k}\{\tau_k\}$ if and only if $r_j \prec r_i$, that is, the consistency in Definition 4 holds.*

Proof. Firstly if $r_j \prec r_i$, then we have $r_i \in \cup_{r_j \preceq r_k}\{r_k\}$, which implies that $\cup_{r_i \preceq r_k}\{r_k\} \subset \cup_{r_j \preceq r_k}\{r_k\}$. So we have that $\cup_{r_j \not\preceq r_k}\{r_k\} \subset \cup_{r_i \not\preceq r_k}\{r_k\}$. In terms of the corresponding relation between r_i and τ_i, we have $\cup_{r_j \not\preceq r_k}\{\tau_k\} \subset \cup_{r_i \not\preceq r_k}\{\tau_k\}$. Conversely, if $\cup_{r_j \not\preceq r_k}\{\tau_k\} \subset \cup_{r_i \not\preceq r_k}\{\tau_k\}$, then we know $\cup_{r_j \not\preceq r_k}\{r_k\} \subset \cup_{r_i \not\preceq r_k}\{r_k\}$. This relation implies $\cup_{r_i \preceq r_k}\{r_k\} \subset \cup_{r_j \preceq r_k}\{r_k\}$. Since $r_i \in \cup_{r_i \preceq r_k}\{r_k\}$, we have $r_j \prec r_i$. Hence, the theorem holds. □

Given a $pk_i = \langle H, V, W_i = [\tau_0]G + [\sum_{r_i \not\preceq r_k} \tau_k]G, \{D_k\}_{\forall r_k \in \uparrow r_i}\rangle$ and a polynomial-time derivation function $F(H, pk_i, r_j) = pk_j$, the relation $pk_i \preceq pk_j$ can be efficiently generated if and only if $r_i \preceq r_j$ in terms of Theorem 2. This gives the consistency between $\mathcal{K}$ and $\mathcal{H}$, that is, $\{pk_i \preceq pk_j\}_{\mathcal{K}} \sim \{r_i \preceq r_j\}_{\mathcal{H}}$.

5.2 Analysis of Correctness

We analyze the validity of our scheme in two cases: $\mathcal{R} = \varnothing$ and $\mathcal{R} \neq \varnothing$ respectively, as follows.

1) *In the Case of $\mathcal{R} = \varnothing$.* By the definition of $\Gamma(r_j, r_i)$, we have the equation

$$\Gamma(r_j, r_i) = \bigcup_{r_j \not\preceq r_l} \{r_l\} \setminus \bigcup_{r_i \not\preceq r_l} \{r_l\} = \bigcup_{r_i \preceq r_l} \{r_l\} \setminus \bigcup_{r_j \preceq r_l} \{r_l\}$$

for $r_i \preceq r_j$. This means that $W_i + \sum_{r_l \in \Gamma(r_j, r_i)} D_l = W_j$ for $W_i = D_0 + \sum_{r_i \not\preceq r_k} D_k$ and $r_i \preceq r_j$, as well as $C_1 + \sum_{r_l \in \Gamma(r_j, r_i)} D'_l = [\zeta_j \cdot \xi]G$. Therefore, the validity of the RBE scheme can be guaranteed by (4).

$$
\begin{aligned}
V' &= e\Big(C_1 + \sum_{r_l \in \Gamma(r_j, r_i)} D'_l, B_{j,k}\Big) \cdot e(A_{j,k}, C_2) \\
&= e\Big([\zeta_j \cdot \xi]G, \Big[\frac{1}{\zeta_j + x'_{j,k}}\Big]H\Big) e\Big(\Big[\frac{x'_{j,k}}{\zeta_j + x'_{j,k}}\Big]G, [\xi]H\Big) \\
&= e(G,H)^{\frac{\zeta_j \cdot \xi}{\zeta_j + x'_{j,k}}} \cdot e(G,H)^{\frac{\xi \cdot x'_{j,k}}{\zeta_j + x'_{j,k}}} \\
&= e(G,H)^{\xi} = V^{\xi}. \qquad (4) \\
V' &= e\Big(C_1 + \sum_{r_l \in \Gamma(r_j, r_i)} D'_l, B^{\mathcal{R}}_{j,k}\Big) \cdot e(A_{j,k}, C_2) \\
&= e\Big([\zeta_j \cdot \xi]G, \Big[\frac{1}{\prod_{l=1}^{t}(\zeta_{i_l} + x'_{i_l,j_l}) \cdot (\zeta_j + x'_{j,k})}\Big]H\Big) \cdot e\Big(\Big[\frac{x'_{j,k}}{\zeta_j + x'_{j,k}}\Big]G, \Big[\frac{\xi}{\prod_{l=1}^{t}(\zeta_{i_l} + x'_{i_l,j_l})}\Big]H\Big) \\
&= e(G,H)^{\frac{\zeta_j \cdot \xi}{(\zeta_j + x'_{j,k}) \prod_{l=1}^{t}(\zeta_{i_l} + x'_{i_l,j_l})}} \cdot e(G,H)^{\frac{\xi \cdot x'_{j,k}}{(\zeta_j + x'_{j,k}) \prod_{l=1}^{t}(\zeta_{i_l} + x'_{i_l,j_l})}} \\
&= e(G,H)^{\frac{\xi}{\prod_{l=1}^{t}(\zeta_{i_l} + x'_{i_l,j_l})}} = (V_{\mathcal{R}})^{\xi}. \qquad (5)
\end{aligned}
$$

2) *In the Case of* $\mathcal{R} \neq \varnothing$. By the definition of $x_{i,j}$ and $x'_{i,j}$, we have

$$\zeta_i + x'_{i,j} = \tau_0 + \sum_{r_i \not\preceq r_k} \tau_k + x'_{i,j} = \tau_0 + x_{i,j},$$

where all $x_{i,j}$ are made public and all $x'_{i,j}$, ζ_i, τ_i are kept secret. Thus, for a revocation set $\mathcal{R} = \{u_{i_l,j_l}, u_{i_k,j_k}\}$ and $i_l \neq i_k$, it is easy to obtain

$$
\begin{aligned}
&\Big[\frac{1}{x_{i_l,j_l} - x_{i_k,j_k}}\Big](B_{i_k,j_k} - B_{i_l,j_l}) \\
&= \Big[\frac{1}{(\zeta_{i_l} + x'_{i_l,j_l})(\zeta_{i_k} + x'_{i_k,j_k})}\Big]H = B_{\mathcal{R}}, \qquad (6)
\end{aligned}
$$

and

$$(V_{i_k,j_k}/V_{i_l,j_l})^{\frac{1}{x_{i_l,j_l} - x_{i_k,j_k}}} = V^{\frac{1}{(\zeta_{i_l} + x'_{i_l,j_l})(\zeta_{i_k} + x'_{i_k,j_k})}} = V_{\mathcal{R}}.$$

Similarly, $B_{\mathcal{R}}$, $B^{\mathcal{R}}_{j,k}$, and $V_{\mathcal{R}}$ can be efficiently computed in an arbitrary revocation set $\mathcal{R}$ by a general recursive method, which is defined in Subsection 5.2.1. Therefore, we can prove (3) by (5).

The revocation mechanism can be supported by (6), that is, for $u_{i,j} \in \mathcal{R}$, $B^{\mathcal{R}}_{j,k}$ cannot be computed because the denominator can be zero in a fraction $\frac{1}{x_{i,j} - x_{i',j'}}$, where $u_{i',j'} \in \mathcal{R}$.

5.2.1 Aggregate Algorithms for User Revocation

It is more important to compute the three values $B^{\mathcal{R}}$, $V^{\mathcal{R}}$, and $B^{\mathcal{R}}_{j,k}$ from the labels of public parameter *par* in an efficient way. We provide such a recursive method (called aggregate algorithm) to solve this problem, as follows.

Given $\mathcal{R} = \{x'_{i_1,j_1}, \ldots, x'_{i_t,j_t}\}$ and their labels $\{lab_{i_k,j_k}\}$ for $k \in [1,t]$ and $lab_{i_k,j_k} = \langle x_{i_k,j_k}, B_{i_k,j_k}, V_{i_k,j_k} \rangle$. In terms of (6), for all $k, l \in [1,t]$, it is easy to obtain the equation

$$
\begin{aligned}
B_{i_k,j_k} - B_{i_l,j_l} &= \Big[\frac{1}{\tau_0 + x_{i_k,j_k}}\Big]H - \Big[\frac{1}{\tau_0 + x_{i_l,j_l}}\Big]H \\
&= \Big[\frac{x_{i_l,j_l} - x_{i_k,j_k}}{(\zeta_{i_l} + x'_{i_l,j_l})(\zeta_{i_k} + x'_{i_k,j_k})}\Big]H.
\end{aligned}
$$

To expand this equation to multi-user cases, we define the following denotation $\widetilde{B}_{s,r}$ for any pair (s,r), where $1 \leqslant s < r \leqslant t$,

$$\widetilde{B}_{s,r} = \Big[\frac{1}{\tau_0 + x_{i_r,j_r}} \cdot \frac{1}{\prod_{k=1}^{s}(\tau_0 + x_{i_k,j_k})}\Big]H.$$

In the same way, we can compute $\widetilde{B}_{s,r} = \Big[\frac{1}{x_{i_r,j_r} - x_{i_s,j_s}}\Big](\widetilde{B}_{s-1,s} - \widetilde{B}_{s-1,r})$. Hence, $B_{\mathcal{R}} = \widetilde{B}_{t-1,t}$ can be completed by computing sequentially $\widetilde{B}_{s,r}$ for $s = [1, t-1]$ and $r = [s+1, t]$ using the equation $(B_{\mathcal{R}} = \widetilde{B}_{t-1,t})$ and the induction

$$
\begin{cases}
\widetilde{B}_{0,r} = B_{i_r,j_r}, & \forall r \in [1,t], \\
\widetilde{B}_{s,r} = \Big[\dfrac{1}{x_{i_r,j_r} - x_{i_s,j_s}}\Big](\widetilde{B}_{s-1,s} - \widetilde{B}_{s-1,r}), & \\
\qquad\qquad s \in [1, t-1],\ r \in [s+1, t], &
\end{cases}
$$

where $\widetilde{B}_{0,r}$ is defined as the initial input B_{i_k,j_k} for $k = [1,r]$. Obviously, we can get $B^{\mathcal{R}}_{j,k}$ in the same way, or it can be computed from the resulting sequence

$(B_{i,j}, \langle \widetilde{B}_{0,1}, \widetilde{B}_{1,2}, \ldots, \widetilde{B}_{t-1,t}\rangle)$, where

$$\begin{cases} \widetilde{B}_{0,t+1} = B_{j,k}, \\ \widetilde{B}_{s,t+1} = \Big[\dfrac{1}{x'_{i_{t+1},j_{t+1}} - x'_{i_s,j_s}}\Big](\widetilde{B}_{s-1,s} - \widetilde{B}_{s-1,t+1}), \\ \qquad \forall s \in [1,t], \\ B^{\mathcal{R}}_{j,k} = \widetilde{B}_{t,t+1}. \end{cases}$$

Similarly, we define $\widetilde{V}_{s,r} = V^{\frac{1}{\tau_0 + x_{i_r,j_r}} \cdot \frac{1}{\prod_{k=1}^{s}(\tau_0 + x_{i_k,j_k})}}$, and then compute $V_{\mathcal{R}}$ from $V_{i_1,j_1}, \ldots, V_{i_t,j_t}$, where $(V_{\mathcal{R}} = \widetilde{V}_{t-1,t})$ and

$$\begin{cases} \widetilde{V}_{0,r} = V_{i_r,j_r}, \qquad \forall r \in [1,t], \\ \widetilde{V}_{s,r} = \Big(\dfrac{\widetilde{V}_{s-1,s}}{\widetilde{V}_{s-1,r}}\Big)^{\frac{1}{x_{i_r,j_r} - x_{i_s,j_s}}}, \\ \qquad \forall s \in [1,t-1], \quad \forall r \in [s+1,t]. \end{cases}$$

5.3 Analysis of Security

We prove the semantic security of our RBE scheme under the assumption of the GDDHE$_1$ problem. Lemma 1 assures that (n,t)-GDDHE$_1$ problem is hard in the generic bilinear groups.

Lemma 1 (Complexity Lower Bound in Generic Bilinear Groups[34]). *Given an (n,t)-GDDHE$_1$ problem, two s-tuples of 3-variate polynomials $F_1, F_2 \in \mathbb{Z}_p[x,y,z]^s$ and a 1-tuple 2-variate polynomial $F_3 \in \mathbb{Z}_p[x,y]$, where $s = n + t + 4$[②], then the maximum total degree of these polynomials is $d = \max(2d_{F_1}, d_{F_2}, d_{F_3}) = \max(2t+4, n+t) \leqslant 2n$. If h is independent of (F_1, F_2, F_3) then for any algorithm $\mathcal{A}$ that makes a total of at most q queries to the oracles computing the group operation in $\mathbb{G}, \mathbb{G}_T$ and the bilinear pairing $e(\cdot,\cdot)$, we have*

$$Adv^{\mathrm{gddhe}}_{\mathcal{A}}(n,t) \leqslant \frac{(q+2(n+t+4)+2)^2 \cdot d}{2p} \leqslant \frac{(q+2(n+t+4)+2)^2 \cdot (2n)}{2p}.$$

In terms of this lemma, we can prove that our RBE scheme is semantically secure against dynamic colluders, the number of which is unlimited, as follows.

Theorem 3. *The (m,n,t)-RBE is semantically secure against dynamic colluders (IND-hcCPA) assuming the (n,t)-GDDHE$_1$ problem is hard in $\mathbb{S}$. Concretely, for any probabilistic algorithm $\mathcal{A}$ that totalizes at most q queries to the oracles performing group operations in $\mathbb{S} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e(\cdot,\cdot))$ and evaluations of the bilinear map $e(\cdot,\cdot)$, we have $Adv^{ind}_{\mathcal{E},\mathcal{A}}(m,n,t) \leqslant \frac{(q+2(n+t+4)+2)^2 \cdot (2n)}{p}$.*

Proof. We prove this theorem according to the IND-hcCPA model as follows: suppose that there exists an adversary $\mathcal{A}$ that can break RBE under collusion attack, we build a reduction algorithm $\mathcal{B}$ to solve above (n,t)-GDDHE$_1$ problem in terms of $\mathcal{A}$.

Given $\mathbb{S} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$, the algorithm $\mathcal{B}$ is given as input an (n,t)-GDDHE$_1$ instance, which is defined by (1). In fact, $\mathcal{B}$ does not know γ, ς but knows $3n$ random integers ζ_i, x_i, x'_i, a_i, $b_i \in \mathbb{Z}^*_p$ in $f(x)$ and $g(x)$, where any pairwise $\{x_i, x'_i\}$ are not equal to each other, but some ζ_i may be equal. Let m be the number of the different ζ_i. In terms of these known values, the algorithm $\mathcal{B}$ will generate an arbitrary role hierarchy $\mathcal{H} = \langle U, R, \preceq\rangle$ with the total number of users n and the number of roles m, and then generates an encryption environment based on $\mathcal{A}$ as follows.

1) *Initial.* $\mathcal{B}$ firstly sets $\overline{G} = [f(\gamma)]G$. Note that $\mathcal{B}$ cannot obtain the value of $\overline{G}$. Then, $\mathcal{B}$ chooses an integer ζ_i from $\{\zeta_i\}_{i\in[1,m]}$ for each role r_i in R, where $i \in [1,m]$. Let $\bar{\zeta}_i = \zeta_i\gamma$ and $W_i = [\bar{\zeta}_i]\overline{G} = [\zeta_i\gamma f(\gamma)]G$ (which can be computed from the input $[\gamma f(\gamma)]G$) for each r_i ($i \in [1,m]$), $\mathcal{B}$ computes the generation matrix $\boldsymbol{M}_{m\times(m+1)}$ reduced from $\mathcal{H}$, and then extends it to $\boldsymbol{M}'_{(m+1)\times(m+1)}$ by appending a row vector $\langle 1,0,\ldots,0\rangle$ in the top of it. So that $\mathcal{B}$ defines the vector $\boldsymbol{W} = \langle W_0, W_1, \ldots, W_m\rangle$ and $\boldsymbol{D} = \langle D_0, \ldots, D_m\rangle$, satisfying (7),

$$\boldsymbol{W} = \boldsymbol{M}' \cdot \boldsymbol{D} = \begin{pmatrix} W_0 \\ W_1 \\ \vdots \\ W_m \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & M_{1,2} & M_{1,3} & \cdots & M_{1,m} \\ \vdots & & & \vdots & \\ 1 & M_{m,2} & M_{m,3} & \cdots & M_{m,m} \end{pmatrix} \cdot \begin{pmatrix} D_0 \\ D_1 \\ \vdots \\ D_m \end{pmatrix} \tag{7}$$

where $W_0 = [\gamma]\overline{G} = [\gamma \cdot f(\gamma)]G$. Obviously, $\mathcal{B}$ can compute $\boldsymbol{D} = (\boldsymbol{M}')^{-1} \cdot \boldsymbol{W}$ and $(\boldsymbol{M}')^{-1} \in \mathbb{Z}_2^{(m+1)\times(m+1)}$ for $rank(\boldsymbol{M}') = m+1$ or a feasible solution $\langle D_0, \ldots, D_m\rangle$ to $\boldsymbol{W} = \boldsymbol{M}' \cdot \boldsymbol{D}$ for $rank(\boldsymbol{M}') \leqslant m$, thus $\mathcal{B}$ can define $\bar{\tau}_i = \tau_i\gamma$ and compute $D_i = [\bar{\tau}_i]\overline{G} = [\tau_i \cdot (\gamma \cdot f(\gamma))]G$ to realize $W_0 = D_0$ and $W_i = D_0 + \sum_{r_i \preceq r_k} D_k$, where $\tau_0 = \gamma$, τ_i can be obtained by $\zeta_i = 1 + \sum_{k=1}^{m} M_{i,k}\,\tau_k$.

②In fact, F_1 and F_2 are $(n+t+4)$-tuples of 3-variate polynomials and F_3 is 1-tuple 2-variate polynomial, such that, $Adv^{gddhe}(t,n,A) \leqslant \frac{(q+(n+t+4)+1+2)^2 \cdot (2n)}{2p}$.

Then it computes easily the public parameter as follows:

$$\begin{cases} \overline{H} = [f(\gamma)g(\gamma)]H = \sum_{i=0}^{n} d_i \cdot [\gamma^i]H, \\ \overline{V} = e(\overline{G}, \overline{H}) = e(G,H)^{f^2(\gamma)g(\gamma)}, \\ \boldsymbol{D} = \langle D_0, D_1, \ldots, D_m \rangle \text{ from } \boldsymbol{M}', \boldsymbol{W}, \end{cases}$$

where $f(x) \cdot g(x) = \sum_{i=0}^{n} d_i \cdot x^i$. Therefore, $\mathcal{B}$ can run $\mathcal{A}$ on these parameters.

2) *Learning.* In this phase, the adversary $\mathcal{A}$ can issue up to t private key queries and $n-t$ label queries to gain the information of this cryptosystem. Let $\mathcal{R} \subset U$ be a subset that indicates at most t corrupted users. Algorithm $\mathcal{B}$ considers three types of queries as follows.

(a) *Hash Query* $(ID, u_{i,j})$. At any time, $\mathcal{A}$ can query the hash function $Hash(ID, u_{i,j})$ and $\mathcal{B}$ replies a random integer in $\mathbb{Z}_p^*$. $\mathcal{B}$ can maintain tables to ensure that repeated queries are answered consistently.

(b) *Private Key Query* $(u_{i,j} \in \mathcal{R})$. $\mathcal{B}$ generates the keys of the corrupted user as follows: for the j-th user in role r_i, $\mathcal{B}$ sets $x'_{i,j} = x_i$ and defines $f_{i,j}(x) = \frac{f(x)}{\zeta_i x + x'_{i,j}} = \prod_{k=1,k\neq i}^{t}(\zeta_k x + x_k)$. Thus for some $e_i \in_R \mathbb{Z}_p$,

$$f_{i,j}(\gamma) = \frac{f(\gamma)}{\bar{\zeta}_i + x'_{i,j}} = \prod_{k=1,k\neq i}^{t} (\zeta_k \gamma + x_k) = \sum_{i=0}^{t-1} e_i \cdot \gamma^i.$$

Since this equation is a polynomial of degree $t-1$, $\mathcal{B}$ can compute

$$\begin{cases} A_{i,j} = \Big[\frac{x'_{i,j}}{\bar{\zeta}_i + x'_{i,j}}\Big]\overline{G} = \Big[\frac{x_i f(\gamma)}{\zeta_i \gamma + x_i}\Big]G \\ \qquad = [x_i \cdot f_{i,j}(\gamma)]G, \\ B_{i,j} = \Big[\frac{1}{\bar{\zeta}_i + x'_{i,j}}\Big]\overline{H} = \Big[\frac{f(\gamma)g(\gamma)}{\zeta_i \gamma + x_i}\Big]H \\ \qquad = [f_{i,j}(\gamma) \cdot g(\gamma)]H, \end{cases}$$

as the decryption key $dk_{i,j} = A_{i,j}$. To generate the labels of users, $\mathcal{B}$ defines $f'_{i,j}(x) = \frac{f(x)}{x + \frac{x_i}{\zeta_i}} = \prod_{k=1,k\neq i}^{t}(x + \frac{x_k}{\zeta_k})$ and $x_{i,j} = \frac{x_i}{\zeta_i} \bmod p$. The value $x_{i,j}$ is stored as $Hash(ID, u_{i,j})$, and $lab_{i,j} = \langle x_{i,j}, V_{i,j}, B_{i,j} \rangle$ can be computed by

$$\begin{aligned} V_{i,j} &= \overline{V}^{\frac{1}{\gamma + x_{i,j}}} = V^{\frac{f^2(\gamma)g(\gamma)}{\gamma + x_i/\zeta_i}} \\ &= e([f'_{i,j}(\gamma)]G, [f(\gamma)g(\gamma)]H). \end{aligned}$$

Finally, $\mathcal{B}$ sends $dk_{i,j}$ and $lab_{i,j}$ to $\mathcal{A}$. Note that, $dk_{i,j}$ is available for the ciphertext which is encrypted by the public encryption key.

(c) *Public Label Query* $(u_{i,j} \notin \mathcal{R})$. If there exists an unused $(\zeta_i x + x'_i)$ in $g(x)$ and $u_{i,j} \in r_i$, $\mathcal{B}$ computes the honest user's label but not their private keys: $\mathcal{B}$ first fixes and records $x_{i,j} = Hash(ID, u_{i,j}) = x'_i/\zeta_i$ and $g'_{i,j}(x) = \frac{g(x)}{x + x'_i/\zeta_i} = \prod_{k=1,k\neq i}^{n-t}(x + \frac{x'_k}{\zeta_k})$, and then computes the user's label $lab_{i,j} = \langle x_{i,j}, V_{i,j}, B_{i,j} \rangle$, where

$$\begin{cases} V_{i,j} = \overline{V}^{\frac{1}{\gamma + x_{i,j}}} = V^{\frac{f^2(\gamma)g(\gamma)}{\gamma + x'_i/\zeta_i}} \\ \qquad = e(G,H)^{f^2(\gamma)g'_{i,j}(\gamma)}, \\ B_{i,j} = \Big[\frac{1}{\gamma + x_{i,j}}\Big]\overline{H} = \Big[\frac{f(\gamma)g(\gamma)}{\gamma + x'_i/\zeta_i}\Big]H \\ \qquad = [f(\gamma) \cdot g'_{i,j}(\gamma)]H, \end{cases}$$

where $f^2(\gamma) \cdot g'_{i,j}(\gamma)$ is the polynomial of degree $n+t-1$. It can be computed because we can obtain $e(G,H)^{\gamma^l}$ for $l \in [0, n+t-1]$ by $[\gamma^l]G$ for $l \in [0, t-1]$ and $[\gamma^l]H$ for $l \in [0, n]$. Finally, $lab_{i,j}$ is given to $\mathcal{A}$.

3) *Challenge.* $\mathcal{A}$ produces two messages $M_0, M_1 \in \mathbb{G}_T$, $r_i \in R$, and returns them to $\mathcal{B}$, where r_i denotes the expected position of encryption. $\mathcal{B}$ picks a random $b \in \{0,1\}$ and constructs a ciphertext $\mathcal{C}_i^b$ as follows:

$$\begin{cases} C_1 = [\varsigma \cdot \bar{\zeta}_i]\overline{G} = [\zeta_i \cdot (\varsigma \cdot \gamma \cdot f(\gamma))]G, \\ C_2 = \Big[\varsigma \cdot \frac{1}{\prod_{i=1}^{t}(\zeta_i \gamma + x_i)}\Big]\overline{H} = \Big[\varsigma \cdot \frac{f(\gamma)g(\gamma)}{f(\gamma)}\Big]H \\ \qquad = [\varsigma \cdot g(\gamma)]H, \\ C_3 = M_b \cdot T, \\ C_4 = \{[\varsigma \cdot \bar{\tau}_k]\overline{G}\}_{\forall r_k \in \uparrow r_i} = \{[\tau_i \cdot (\varsigma \cdot \gamma \cdot f(\gamma))]G\}_{\forall r_k \in \uparrow r_i}. \end{cases}$$

Finally, $\mathcal{B}$ sends the challenge ciphertext $\mathcal{C}_i^b$ to $\mathcal{A}$.

4) *Guess.* $\mathcal{A}$ returns $b' \in \{0,1\}$. If $b = b'$, $\mathcal{B}$ outputs 1 (True), otherwise 0 (False).

This completes the description of algorithm $\mathcal{B}$. It is easy to describe the advantage of adversary in both instances. To do so, we recall two polynomials

$$\begin{aligned} f(x) &= \prod_{i=1}^{t}(\zeta_i x + x_i) = \prod_{i=1}^{t}\Big(x + \frac{x_i}{\zeta_i}\Big) \\ g(x) &= \prod_{i=1}^{n-t}(\zeta_{t+i} x + x'_i) = \prod_{i=1}^{n-t}\Big(x + \frac{x'_i}{\zeta_{t+i}}\Big) \end{aligned}$$

where $\prod_{i=1}^{t}\zeta_i = \prod_{i=1}^{n-t}\zeta_{t+i} = 1 \pmod p$. If T is equal to $e(G,H)^{\varsigma f(\gamma)g(\gamma)}$, the challenge ciphertext $\mathcal{C}_i$ is available since

$$\begin{aligned} C_3 &= M_b \cdot e(\overline{G}, \overline{H})^{\frac{\varsigma}{\prod_{i=1}^{t}(\zeta_i \gamma + x_i)}} \\ &= M_b \cdot e(G,H)^{\varsigma \cdot \frac{f^2(\gamma)g(\gamma)}{f(\gamma)}} = M_b \cdot T. \end{aligned}$$

Hence, we have

$$\begin{aligned} &\Pr[b = b' : T \leftarrow e(G,H)^{\varsigma f(\gamma)g(\gamma)}] \\ &\qquad = \Pr[\mathcal{A}(\mathcal{C}_i^b) = b : T \leftarrow e(G,H)^{\varsigma f(\gamma)g(\gamma)}]. \end{aligned}$$

Otherwise, we have $\Pr[b = b' | T \leftarrow_R \mathbb{G}_T] = \Pr[b \neq b' | T \leftarrow_R \mathbb{G}_T] = 1/2$ and the equation[③]

$$\begin{aligned}
&Adv_{\mathcal{B}}^{\text{gddhe}}(n,t)\\
&= \left| \begin{array}{l} \Pr[\mathcal{B}((F_1,F_2,F_3,T)] = 1 \\ -\Pr[\mathcal{B}((F_1,F_2,F_3,T)] = 0 \end{array} \right|\\
&= |\Pr[b = b'] - \Pr[b \neq b']|\\
&= \left| \begin{array}{c} \Pr[b = b' : T \leftarrow e(G,H)^{\varsigma f(\gamma)g(\gamma)}] \\ -\Pr[b \neq b' : T \leftarrow_R \mathbb{G}_T] \end{array} \right|\\
&= |\Pr[\mathcal{A}(\mathcal{C}_i^b) = b : T \leftarrow e(G,H)^{\varsigma f(\gamma)g(\gamma)}] - 1/2|.
\end{aligned}$$

Therefore, according to Subsection 2.2, we have

$$\begin{aligned}
&Adv_{\mathcal{E},\mathcal{A}}^{\text{ind}}(m,n,t)\\
&= \left| \begin{array}{l} \Pr[b = b' : T \leftarrow e(G,H)^{\varsigma f(\gamma)g(\gamma)}] - \\ \Pr[b \neq b' : T \leftarrow e(G,H)^{\varsigma f(\gamma)g(\gamma)}] \end{array} \right|\\
&= |2\Pr[b = b' : T \leftarrow e(G,H)^{\varsigma f(\gamma)g(\gamma)}] - 1|\\
&= 2|\Pr[\mathcal{A}(\mathcal{C}_i) = b : T \leftarrow e(G,H)^{\varsigma f(\gamma)g(\gamma)}] - 1/2|.
\end{aligned}$$

Summing up, we get that $Adv_{\mathcal{B}}^{\text{gddhe}}(n,t) = Adv_{\mathcal{E},\mathcal{A}}^{\text{ind}}(m,n,t)/2$. In terms of Lemma 1, it is easy to see that $Adv_{\mathcal{E},\mathcal{A}}^{\text{ind}}(m,n,t) \leqslant \frac{(q+2(n+t+4)+2)^2 \cdot (2n)}{p}$. This implies the algorithm can decide GDDHE_1 problem with a non-negligible success probability, which would contradict with the assumption. Moreover, we can prove the same result when $t = n$, that is, full collusion security. □

6 Performance Analysis

Following our terminology, we denote $|R| = m$ and $|\mathcal{R}| = t$. We use E to denote a multiplication operation in $\mathbb{G}_1, \mathbb{G}_2$ or an exponentiation operation in $\mathbb{G}_T$, P to denote the pairing operation $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. We analyze the performance of each phase in our scheme as shown in the second column of Table 3. We neglect the hash operation, the operations in $\mathbb{Z}_p$, an addition in $\mathbb{G}_1, \mathbb{G}_2$ and a multiplication in $\mathbb{G}_T$, since these operations are much more efficient compared to the pairing operation and exponentiation. Considering the revocation mechanism, our scheme requires $\frac{1}{2}t(t+1)\big(E(\mathbb{G}_2)+E(\mathbb{G}_T)\big)$ in the encryption algorithm, and $(2t+1)E(\mathbb{G}_2)$ is required in the decryption algorithm. This indicates that our scheme has a low computation overhead, related to the number of granted roles and revoked users.

We also present the communication complexity in the third column of Table 3. This indicates that our scheme has a short constant-size private user key even in large scale systems. Moreover, the size of ciphertext is $O(m+t)$, which is proportional to the number of granted roles and revoked users. Hence, our construction achieves the optimal bound of overhead rate for both ciphertexts and decryption keys.

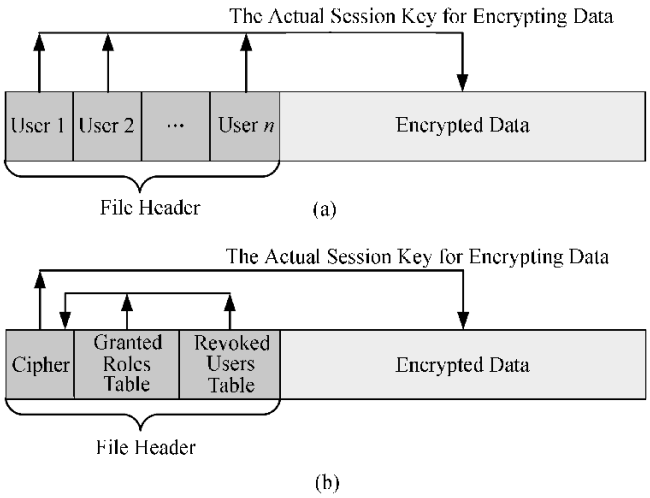

Fig.2. Comparison between exiting encrypted file system in Windows NT (a) and our scheme (b).

Our scheme can construct an efficient encrypted file systems (EFS) based on RBAC, which enables the users to encrypt files on disks in terms of the user's role(s). Many existing encrypted file systems implement the straight forward encryption system where the number of ciphertexts in the file header grows linearly in the number of users that can access the file. As a result, there is often a hard limit on the number of users that can access a file, and the headers of all files must be changed to permit the user's access when a new user joins the system. For example, the following quote is from Microsoft's knowledge base: "EFS has a limit of 256 KB in the file header for the EFS metadata. This limits the number of individual entries for file sharing that may be added. On average, a maximum of 800 individual users may be added to an encrypted file."[37] We show such a structure in Fig.2(a).

However, the RBAC systems built on our RBE scheme can automatically use the role key to encrypt the files in terms of the user's role r_i in a transparent way for users. Such a file header is shown in Fig.2(b), in which "Cipher" consists of the constant-size $C_1, C_2, C_3 \in \mathcal{C}_i$, "Granted roles table" consists of $C_4 \in \mathcal{C}_i$, and "Revoked users table" consists of the list

③Let R and Z denote $T \leftarrow_R \mathbb{G}_T$ and $T \leftarrow e(G,H)^{\varsigma f(\gamma)g(\gamma)}$, respectively. It is easy to show that

$$\begin{aligned}
&|\Pr[b = b'] - \Pr[b \neq b']| = |\Pr[b = b' : R]\Pr[R] + \Pr[b = b' : Z]\Pr[Z] - \Pr[b \neq b' : R]\Pr[R] - \Pr[b \neq b' : Z]\Pr[Z]|\\
&= |\frac{1}{2}(2\Pr[b = b' : Z] - 1 + 1 - 2\Pr[b \neq b' : R])| = |\Pr[b = b' : Z] - \Pr[b \neq b' : R]|.
\end{aligned}$$

Table 3. Performance Analysis for RBE

	Computation Complexity	Communication Complexity
Setup (Public parameter)	$(m+1)E(\mathbb{G}_1)+1\mathrm{P}$	$(m+1)\mathbb{G}_1+1\mathbb{G}_2+1\mathbb{G}_T$
GenRKey (Public role key)		$1\mathbb{Z}_p^*+m\mathbb{G}_1+1\mathbb{G}_2+1\mathbb{G}_T$
AddUser (Private user key)	$1E(\mathbb{G}_T)$	$1\mathbb{Z}_p^*+1\mathbb{G}_2+1\mathbb{G}_T$
Encrypt (Ciphertext)	$(m+1)E(\mathbb{G}_1)+1E(\mathbb{G}_2)+1E(\mathbb{G}_T)$	$(m+1)\mathbb{G}_1+1\mathbb{G}_2+1\mathbb{G}_T$
Decrypt	$2P$	

of user's labels in $\mathcal{R} \in \mathcal{C}_i$ for an RBE ciphertext $\mathcal{C}_i$. Here, both the number of users and the number of revoked users are not limited in this EFS system. Moreover, for a new user who joins this system, all existing files need not be changed to permit the access of these files.

To reduce the length of list of revoked users in file header, our RBE scheme has the capability of fixing the user's labels into the public role key by using the Aggregate algorithms in Subsection 5.2, such that these users will be permanently revoked. Furthermore, as a practical cryptosystem, the keys in RBE system should be regularly renewed by the system manager (such as 3 months or half year). The users who leave the system should be permanently revoked after updating keys. Hence, these mechanisms can avoid the accumulation of revoked users in RBE systems, as well as can reduce the length of $\mathcal{R}$ in file headers.

For the sake of clarity, we evaluate the performance of EFS on our RBE scheme as follows: suppose the security parameter κ is 80-bits[38-39], we need the elliptic curve domain parameters over $\mathbb{Z}_p$ with $|p| =$ 160-bits. Elliptic curve domain parameters over $\mathbb{Z}_p$ with $\lceil \log_2 p \rceil = 2\kappa$ supply approximately κ bits of security[40], which means that solving the logarithm problem on associated elliptic curve is believed to take approximately 2^κ operations. This means that the length of the integer is $l_0 = 2\kappa$ in $\mathbb{Z}_p$. Similarly, we have the length of the element in $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$, which satisfy $l_1 = 4\kappa$, $l_2 = 20\kappa$, and $l_T = 10\kappa$. We assume that the embedding degree of elliptic curve is 5. In the RBE scheme, the length of the file header is $(m+1)\cdot l_1+1\cdot l_2+1\cdot l_T+t\cdot l = 4\kappa\cdot(m+1)+20\kappa+10\kappa+128t = 320m+128t+2720$ bits, where l is the length of user's label and is set to 128 bits presumed. Considering a system where each role contains 40 users on average, with 800 users and 100 revoked users, the file header is just $320\times 20+128\times 100+2720 = 21920$ bit ≈ 2.68 KB. In contract to the existing EFS structure of Windows NT[41], this storage cost is far less than 256 KB, which is the exact size of file header of encrypted files in Windows EFS. Furthermore, in RBE-based EFS, the file header with 256 KB can support the system with about 2000 roles, and each role contains 300 users on average and 11 000 revoked users, where the length of the user label is 128 bit. In theory, the above-mentioned system can support unlimited number of users, which is much better than existing EFS.

7 Conclusion and Future Work

In this paper, we introduced a generic role-based encryption over RBAC model to support a flexible encryption of resources in RBAC systems. The proposed scheme supports fully collusion security under a special case of the GDDHE problem and implements the revocation at minimal cost and constant-size ciphertexts and decryption keys. Our scheme has better performance and scalability than existing solutions in encrypted file systems.

In our future work, we will investigate a more comprehensive role-based cryptosystem to support various secure mechanisms, such as encryption, signature, and authentication. Meanwhile, we would exploit the partial ordering relation in ABE with respect to the work addressed in this paper. We will also optimize our solution to improve the performance of revocation algorithms in our scheme. Finally, based on our exiting work, we will propose a complete cryptosystem to realize massive-scale conditional access systems for the practical RBAC applications of large-scale organizations.

References

[1] Sandhu R, Ferraiolo D F, Kuhn D R. The nist model for role-based access control: Towards a unified standard. In *Proc. the 5th ACM Workshop on Role Based Access Control* (*RBAC*), Berlin, Germany, Jul. 26-27, 2000, pp.47-63.

[2] Li Q, Zhang X W, Xu M W, Wu J P. Towards secure dynamic collaborations with group-based RBAC model. *Computers & Security*, 2009, 28(5): 260-275.

[3] Shafiq B, Joshi J, Bertino E, Ghafoor A. Secure interoperation in a multidomain environment employing RBAC policies. *IEEE Transactions on Knowledge and Data Engineering*, 2005, 17(11): 1557-1577.

[4] Zhu Y, Ahn G J, Hu H X, Wang H X. Cryptographic role-based security mechanisms based on role-key hierarchy. In *Proc. the 5th ACM Symposium on Information, Computer and Communications Security* (*ASIACCS*), Beijing, China, Apr. 13-16, 2010, pp.314-319.

[5] Akl S G, Taylor P D. Cryptographic solution to a problem of access control in a hierarchy. *ACM Transactions on*

Computer System, 1983, 1(3): 239-248.

[6] Akl S G, Taylor P D. Cryptographic solution to a multilevel security problem. In *Proc. Advances in Cryptology: CRYPTO*, Santa Barbara, USA, 1982, pp.237-249.

[7] Wallner D M, Harder E G, Agee R C. Key management for multicast: Issues and architecture. Internet Draft, draft-waller-key-arch-01.txt, 1998.

[8] Wong C K, Gouda M, Lam S S. Secure group communications using key graphs. In *Proc. the Annual Conference of the Association for Computing Machinery's Special Interest Group on Data Communication* (*SIGCOMM*), Vancouver, Canada, Sept. 2-4, 1998, 28, pp.68-79.

[9] Asano T. Reducing receiver's storage in CS, SD and LSD broadcast encryption schemes. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 2005, 88(1): 203-210.

[10] Naor D, Naor M, Lotspiech J. Revocation and tracing schemes for stateless receivers. In *Proc. the 21st Annual International Cryptology Conference* (*CRYPTO*), Santa Barbara, USA, Aug. 19-23, 2001, pp.41-62.

[11] Halevy D, Shamir A. The LSD broadcast encryption scheme. In *Proc. the 22nd International Cryptology Conference* (*Crypto*), Santa Barbara, USA, Aug. 18-22, 2002, pp.47-60.

[12] Boneh D, Franklin M. Identity-based encryption from the weil pairing. In *Proc. the 21st Annual International Cryptology Conference* (*CRYPTO*), Santa Barbara, USA, Aug. 19-23, 2001, pp.213-229.

[13] Yuen T H, Susilo W, Mu Y. How to construct identity-based signatures without the key escrow problem. *International Journal of Information Security*, 2010, 9(4): 297-311.

[14] Gentry C, Silverberg A. Hierarchical ID based cryptography. In *Proc. the 8th International Conference on the Theory and Application of Cryptology and Information Security* (*ASIACRYPT*), Queenstown, New Zealand, Dec. 1-5, 2002, pp.548-566.

[15] Tzeng W G. A time-bound cryptographic key assignment scheme for access control in a hierarchy. *IEEE Transactions on Knowledge and Data Engineering*, 2002, 14(1): 182-188.

[16] Sahai A, Waters B. Fuzzy identity-based encryption. In *Proc. the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques* (*EUROCRYPT*), Aarhus, Denmark, May 22-26, 2005, pp.457-473.

[17] Goyal V, Pandey O, Sahai A, Waters B. Attribute-based encryption for fine-grained access control of encrypted data. In *Proc. the 13th ACM Conference on Computer and Communications Security* (*CCS*), Alexandria, USA, Oct. 30-Nov. 3, 2006, pp.89-98.

[18] Ostrovsky R, Sahai A, Waters B. Attribute-based encryption with non-monotonic access structures. In *Proc. the 14th ACM Conference on Computer and Communications Security* (*CCS*), Alexandria, USA, Oct. 28-31, 2007, pp.195-203.

[19] Chase M. Multi-authority attribute based encryption. In *Proc. the 4th Theory of Cryptography Conference* (*TCC*), Amsterdam, The Netherlands, Feb. 21-24, 2007, pp.515-534.

[20] Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption. In *Proc. 2007 IEEE Symposium on Security and Privacy* (*S&P*), Oakland, USA, May 20-23, 2007, pp.321-334.

[21] Waters B. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. Cryptology ePrint Archive, Report 2008/290, 2008, http://eprint.iacr.org/.

[22] Goyal V, Jain A, Pandey O, Sahai A. Bounded ciphertext policy attribute based encryption. In *Proc. the 35th International Colloquium on Automata, Languages and Programming, Part II — Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations* (*ICALP*(2)), Reykjavik, Iceland, Jul. 7-11, 2008, pp.579-591.

[23] Ibraimi L, Tang Q, Hartel P H, Jonker W. Efficient and provable secure ciphertext-policy attribute-based encryption schemes. In *Proc. the 5th International Conference on Information Security Practice and Experience* (*ISPEC*), Xi'an, China, Apr. 13-15, 2009, pp.1-12.

[24] Attrapadung N, Imai H. Dual-policy attribute based encryption. In *Proc. the 7th International Conference on Applied Cryptography and Network Security* (*ACNS*), Paris, France, Jun. 2-5, 2009, pp.168-185.

[25] Attrapadung N, Imai H. Dual-policy attribute based encryption: Simultaneous access control with ciphertext and key policies. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 2010, E93-A(1): 116-125.

[26] Wang L Y, Wijesekera D, Jajodia S. A logic-based framework for attribute based access control. In *Proc. the 2004 ACM Workshop on Formal Methods in Security Engineering* (*FMSE*), Washington DC, USA, Oct. 29, 2004, pp.45-55.

[27] Frikken K B, Atallah M J, Li J T. Attribute-based access control with hidden policies and hidden credentials. *IEEE Transaction on Computers*, 2006, 55(10): 1259-1270.

[28] Schoinas I, Falsafi B, Lebeck A R, Reinhardt S K, Larus J R, Wood D A. Fine-grain access control for distributed shared memory. In *Proc. the 6th International Conference on Architectural Support for Programming Languages and Operating Systems* (*ASPLOS*), San Jose, USA, Oct. 4-7, 1994, pp.297-306.

[29] Damiani E, Vimercati S D C D, Paraboschi S, Samarati P. A fine-grained access control system for xml documents. *ACM Transactions on Information and System Security*, 2002, 5(2): 169-202.

[30] Shahandashti S F, Naini R S. Threshold attribute-based signatures and their application to anonymous credential systems. In *Proc. the 2nd International Conference on Cryptology in Africa* (*AFRICACRYPT*), Gammarth, Tunisia, Jun. 21-25, 2009, pp.198-216.

[31] Maji H, Prabhakaran M, Rosulek M. Attribute-based signatures: Achieving attribute-privacy and collusion-resistance. Cryptology ePrint Archive, Report 2008/328, 2008, http://eprint.iacr.org/.

[32] Wang H X, Zhu Y, Feng R Q. Attribute-based signature with policy-and-endorsement mechanism. *Journal of Computer Science and Technology*, 2010, 25(6): 1293-1304.

[33] Attrapadung N, Imai H. Attribute-based encryption supporting direct/indirect revocation modes. In *Proc. the 12th IMA International Conference on Cryptography and Coding*, Cirencester, UK, Dec. 15-17, 2009, pp.278-300.

[34] Boneh D, Boyen X, Goh E J. Hierarchical identity based encryption with constant size ciphertext. In *Proc. the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques* (*EUROCRYPT*), Aarhus, Denmark, May 22-26, 2005, pp.440-456.

[35] Boneh D, Gentry C, Waters B. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *Proc. the 25th Annual International Cryptology Conference* (*CRYPTO*), Santa Barbara, USA, Aug. 14-18, 2005, pp.258-275.

[36] Toahchoodee M, Xie X, Ray I. Towards trustworthy delegation in role-based access control model. In *Proc. the 12th International Conference on Information Security* (*ISC*), Pisa, Italy, Sept. 7-9, 2009, pp.379-394.

[37] Microsoft Corporation. How encrypting file system works. Microsoft TechNet Report, 2009, http://technet.microsoft.com/en-us/library/cc781588(WS.10).aspx.

[38] SEC1. Standards for efficient cryptograhy group: Elliptic curve cryptography, Version 1.0, 2000.

[39] SEC2. Standards for efficient cryptograhy group: Recommended elliptic curve domain parameters, Version 1.0, 2000.

[40] Su D, Lv K W. A new hard-core predicate of paillier's trapdoor function. In *Proc. the 10th International Conference on Cryptology in India* (*INDOCRYPT*), New Delhi, India, Dec. 13-16, 2009, pp.263-271.

[41] Schultz E E. Windows 2000 security: A postmortem analysis. *Network Security*, 2004, 2004(1): 6-9.

Yan Zhu is an associate professor of computer science in the Institute of Computer Science and Technology at Peking University since 2007. He worked at the Department of Computer Science and Engineering, Arizona State University as a visiting associate professor from 2008 to 2009. His research interests include cryptography and network security.

Hong-Xin Hu is currently working toward the Ph.D. degree at the School of Computing, Informatics and Decision Systems Engineering, Ira A. Fulton School of Engineering, Arizona State University, Tempe. He is also a member of the Security Engineering for Future Computing Laboratory, Arizona State University. His current research interests include access control models and mechanisms, security in social network and cloud computing, network and distributed system security and secure software engineering.

Gail-Joon Ahn received the Ph.D. degree in information technology from George Mason University, Fairfax, USA, in 2000. He was an associate professor at the College of Computing and Informatics, and the Founding Director of the Center for Digital Identity and Cyber Defense Research and Laboratory of Information Integration, Security, and Privacy, University of North Carolina, Charlotte. He is currently an associate professor in the School of Computing, Informatics, and Decision Systems Engineering, Ira A. Fulton School of Engineering and the Director of Security Engineering for Future Computing Laboratory, Arizona State University, Tempe. His research interests include information and systems security, vulnerability and risk management, access control, and security architecture for distributed systems, which has been supported by the U.S. National Science Foundation, National Security Agency, U.S. Department of Defense, U.S. Department of Energy, Bank of America, Hewlett Packard, Microsoft, and Robert Wood Johnson Foundation. He is a recipient of the U.S. Department of Energy CAREER Award and the Educator of the Year Award from the Federal Information Systems Security Educators Association.

Huai-Xi Wang received his B.S. degree from the School of Mathematical Sciences, Peking University in 2006. He is a Ph.D. candidate at Peking University. His research interests include attribute based system and pairing based cryptography.

Shan-Biao Wang received his B.S. degree from the School of Mathematical Sciences, Peking University in 2007. He is a Ph.D. candidate at Peking University. His research interests include secure computation and lattice based cryptography.