Role-Based Cryptosystem: A New Cryptographic RBAC System Based on Role-Key Hierarchy

Yan Zhu, Gail-Joon Ahn, *Senior Member, IEEE*, Hongxin Hu, Di Ma, Shanbiao Wang

***Abstract*—Even though role-based access control (RBAC) can tremendously help us minimize the complexity in administering users, it still needs to realize the notion of roles at the resource level. In this paper, we propose a practical cryptographic RBAC model, called role-key hierarchy model, to support various security features including signature, identification and encryption on role-key hierarchy. In addition, several advanced features, such as role or user revocation, tracing, and anonymity, are implemented as well. With the help of rich algebraic structure of elliptic curves, we introduce a unified and complete construction of role-based cryptosystem to verify the rationality and validity of our proposed model. Also, a proof-of-concept prototype implementation and performance evaluation are discussed to demonstrate the feasibility and efficiency of our mechanisms.**

***Index Terms*—Security, access control, role-based cryptosystem, role-key hierarchy, role and user revocation.**

I. Background and Motivation

ROLE-BASED access control (RBAC), as a proven alternative to traditional access control approaches including discretionary access control (DAC) and mandatory access control (MAC), has been widely adopted for various information systems over the past few years [1]. Even though RBAC can tremendously help us minimize the complexity in administering users, it still needs to realize the notion of roles at the resource level. For example, RBAC provides an effective protection for the resources in systems, but such a protection will be invalid if the resources break away from the systems. Thus, RBAC systems need to control a user's access to resources as well as resource-level management based on roles. In order to provide effective resource management, it is inevitable to adopt various cryptographic capabilities for managing resources in RBAC systems. However, 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 due to inconsistent parameters among cryptographic mechanisms.

In distributed environments, we can leverage RBAC models to enforce fine-grained policies for sharing resources [2]. However, current cryptosystems do not support such shared modes because cryptographic keys cannot be recognized between different RBAC systems. Consequently, resources have to be re-encrypted when they are transferred into another domain. So, it is necessary to design an efficient cryptographic mechanism compatible with corresponding access control systems.

Related Work. The research for cryptographic hierarchical structure 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 structure have been proposed. Akl and Taylor introduced a simple scheme to solve multilevel security problem [3], [4]. Since then, several efficient methods have been studied. The concept of Logical Key Hierarchy (LKH) was proposed by Wallner et al. [5] and Wong et al. [6]. In this paradigm, common encryption keys were organized into a tree structure to achieve secure group communication in the multicast environment. Additionally, public-key hierarchy cryptosystems have been recently proposed. For instance, hierarchical identity-based encryption (HIBE) mirrors an organizational hierarchy [7]. Although the public key can be an arbitrary multi-level string, the HIBE schemes support for tree structures (but not for inverse-tree structures and general hierarchies, which provide the aggregation of resources) and provide an efficient method to assign a subset of users to decrypt the message.

Another important area is hierarchical key management (HKM) that also organizes the keys into a hierarchy. For example, time-bound hierarchical key assignment (THKA) [8] can assign time-dependent encryption keys to a set of classes in a partially ordered hierarchy. This scheme is especially suited for realtime broadcast systems with time control. Unfortunately, these existing schemes are group-oriented and awkward to handle individual keys because all users with the same identity (or security level) share the same key. Therefore, we attempt to construct an effective scheme that is group-oriented with a hierarchical structure, and also has

Manuscript received January 28, 2013; revised June 14, 2013; accepted August 30, 2013; date of publication October 8, 2013. Date of current version October 24, 2013. The work of Y. Zhu and S. Wang was supported by the National Natural Science Foundation of China (Project No. 61170264 and No. 10990011) and the National 973 Program (Project No. 2013CB329606). The work of G.-J. Ahn and H. Hu was partially supported by the grants from US National Science Foundation and US Department of Energy. This work was presented in part at AsiaCCS'10, Beijing, China, 2010. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. C.-C. Jay Kuo.

Y. Zhu is with the School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China (e-mail: zhuyan@ustb.edu.cn).

G.-J. Ahn is with the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, Arizona, 85287 (e-mail: gahn@asu.edu).

H. Hu is with the Department of Computer and Information Sciences, Delaware State University, Dover, DE 19901 (e-mail: hhu@desu.edu).

D. Ma is with the Department of Computer and Information Science, University of Michigan-Dearborn MI 48128 (e-mail: dmadma@umich.edu).

S. Wang is with the School of Mathematics Science, Peking University, Beijing 100871, China. (e-mail: shanbiaowang@pku.edu.cn).

the flexibility to handle individuals, such that it is able to realize several advanced security functions such as revocation, undeniability and traceability.

Several new technologies, such as identity-based encryption (IBE) [9], attribute-based encryption (ABE) [10], and public-key broadcast encryption (PBE) [11], lay out a solid foundation for designing an efficient cryptosystem. Inspired by these techniques, we have proposed a cryptographic RBAC model [12] that introduces a hierarchical role-based access control model into public key cryptosystem. Hereafter, based on such a model, several role-based encryption (RBE) schemes have been proposed for secure data storage, such as [13] and [14]. The former [13] introduced a revocation mechanism into the encryption process, supporting both role and user revocations for any number of roles and users. The latter [14] paid more attention to design a scheme for storing data securely in the cloud environment, as well as providing user revocation support.

Contribution. In this paper, we propose a more practical cryptographic RBAC model based on [12], called role-key hierarchy model, to support a variety of security features including signature, identification and encryption based on role-key hierarchy. With the help of rich algebraic structure of elliptic curve, we introduce a role-based cryptosystem construction to verify the rationality and validity of our proposed model. This construction provides more efficient and flexible control than other hierarchical key assignments [15]. More importantly, some unique security mechanisms, such as role-based signature, authentication, and encryption, are supported by our construction. In addition, several advanced features, such as role or user revocation, tracing, and anonymous, could be implemented as well.

Organization. The rest of the paper is organized as follows. Section II overviews the role hierarchy in RBAC and Section III articulates our role-key hierarchy structure along with the usability of this structure in Section IV. In Section V and VI, we address our RBC construction and application schemes in depth. In Section VII, we evaluate the security and performance of our schemes. Finally, we conclude this paper with our future work.

II. Preliminaries

A. Partial Orders

Let $\Psi = \langle P, \preceq \rangle$ be a (finite) partially ordered set with partial order relation $\preceq$ on a (finite) set P. A partial order is a reflexive, transitive and anti-symmetric binary relation. Inheritance is reflexive because a role inherits its own permissions, transitivity is a natural requirement in this context, and anti-symmetry rules out roles that inherit from one another and would therefore be redundant.

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 inequality: $x \prec y$ in P if and only if (or iff) $x \preceq y$ and $x \neq y$. Also, if x is dominated by y, we denote the domination relation as $x \prec_d y$. In addition, if $x \prec_d y$ and $x \preceq z \prec y$, it then implies $z = x$. The latter condition demands that there be no element z of P satisfying $x \prec z \prec y$. 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.

B. Role Hierarchy

In an information system, a hierarchy is used to denote the relationships and arrangements of objects, users, elements, values, and so on. Especially, in many access control systems the users are organized in a hierarchy constructed with a number of classes, called security classes or roles, according to their competencies and responsibilities. This hierarchy arises from the fact that some users have more access rights than others, hence it has been widely adopted by most access control models, including RBAC.

In order to manage large-scale systems, the hierarchy in RBAC becomes more complex than other systems. Especially, role hierarchy (RH) is a natural means for structuring roles to reflect an organization's lines of authority and responsibility. We adopt the definitions from RBAC models proposed by Sandu et al. [16]:

Definition 1: [Hierarchical RBAC model]: The $RBAC$ model has the following components:

- **U**, **R**, **P**, and **S**, users, roles, permissions and sessions respectively,
- **PA** $\subseteq P \times R$, a many-to-many permission to role assignment relation.
- **UA** $\subseteq U \times R$, a many-to-many user to role assignment relation.
- **RH** $\subseteq R \times R$ is a partial order on R called the *role hierarchy* or role dominance relation, written as $\preceq$,
- **user** : $S \rightarrow U$, a function mapping each session s_i to the single user $user(s_i)$, and
- **roles** : $S \rightarrow 2^R$, a function mapping each session s_i to a set of roles: $roles(s_i) \subseteq \{r \in R | \exists r' \in R, r \preceq r' : (user(s_i), r') \in UA\}$ and s_i has the permissions: $\bigcup_{r \in roles(s_i)} \{p \in P | \exists r'' \in R, r'' \preceq r : (p, r'') \in PA\}$.

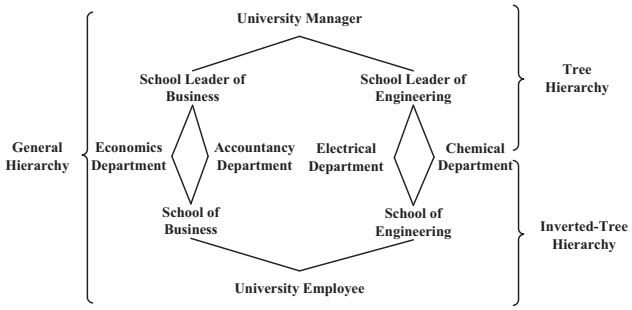


Fig. 1. Example of role hierarchy with tree, inverted-tree, and general hierarchies.

A hierarchy in RBAC is mathematically a partial order that defines an inheritance (or seniority) relation between roles, whereby senior roles acquire the permissions of their juniors. An example of RH is shown in Fig. 1, in which more powerful (senior) roles are shown toward the top of the

diagram and less powerful (junior) roles toward the bottom. Based on the specific features of resource management, we divide RH into three categories: tree, inverted-tree, and general hierarchy (which composes various different structures into a role hierarchy).

III. Role Key Hierarchy

A. Role-Key Hierarchy Structure

In order to incorporate cryptographic schemes with RBAC, we propose a new hierarchy structure called **Role-Key Hierarchy** (RKH). Based on the hierarchical RBAC model, we define RKH as follows:

Definition 2: [Role-Key Hierarchy]: Given a role hierarchy $\langle R, \preceq \rangle$ in RBAC, role-key hierarchy, denoted by $\mathcal{H} = \langle U, K, R, P, \preceq \rangle$, is a cryptographic partial order relation for the set of keys based on $\langle R, \preceq \rangle$, users ($U$), roles ($R$), and permissions ($P$), satisfying the following conditions:

1) $\mathbf{K} = \mathbf{PK} \cup \mathbf{SK} \cup \mathbf{AG}$, the key set K includes the role-key set PK, the user-key set SK, and the algorithm set AG;
2) $\mathbf{UKA} \subseteq U \times SK$, a one-to-one user to key assignment relation, i.e., each user $u_{i,j} \in U$ is assigned to an exclusive *user-key* $sk_{i,j} \in SK$;
3) $\mathbf{RKA} \subseteq R \times PK$, a one-to-one role to key assignment relation, i.e., each role $r_i \in R$ corresponds to a unique *role-key* $pk_i \in PK$;
4) $\mathbf{PKA} \subseteq P \times (PK \times SK \times AG)$, a many-to-many permission to key assignment relation, i.e., each permission $p \in P$ corresponds to a set of triples $(sk_{i,j}, pk_l, A_p) \in (SK, PK, AG)$;
5) $\mathbf{KH} \subseteq PK \times PK$, a partial order on PK called the key hierarchy or key dominance relation, also written as $\preceq$; and
6) $\mathbf{keys} : S \rightarrow 2^K$, a function mapping each session s_i to a set of role keys, $keys(s_i) \subseteq \{pk_l \in PK | \exists r, r' \in R, r' \preceq r : (user(s_i), r) \in UA, (r', pk_l) \in RKA\}$ and there is an algorithm $A_p \in AG$ that can realize $p \in \bigcup_{r \in roles(s_i)} \{p \in P | \exists r'' \in R, r'' \preceq r : (p, r'') \in PA\}$ for the key pair $(sk_{j,k}, pk_l) \in SK \times PK$, where $(user(s_i), sk_{j,k}) \in UKA$, $pk_l \in keys(s_i)$, and $(p, (sk_{j,k}, pk_l, A_p)) \in PKA$.

where, $\langle K, \preceq \rangle$ is the smallest partially ordered set satisfying the above conditions, and in $u_{i,j}, sk_{i,j}$, i and j represent the index variable of role and user, respectively. A user holds multiple user keys if he is a member of multiple roles in the role hierarchy.

In this definition, condition 6) means that each user $u_{i,j}$ can access the resources associated with r_l if and only if $r_l \preceq r_i \in RH$ and $(u_{i,j}, r_i) \in UA$. In RBAC systems, various access control functions are designated by permissions P. In the same way, the RBAC permissions can be designated by some cryptographical algorithms, such as $Encrypt$ and $Decrypt$, which can realize various access control functions by using role keys and user keys in role-key hierarchy. These algorithms can also be used independently to protect files from unauthorized access while these resources break away from the scope of this RBAC systems or an attacker gains physical access to the computer.

For the sake of clarity, we show the structure model of a new hierarchical RBAC model with role-key hierarchy in Fig. 2. By using three map functions, UKA, RKA, PKA, three entities, (U, R, P), in original RBAC model correspond to three entities (SK, PK, AG) of the key space K, respectively. Moreover, a map function $keys$ implements the mapping between each session to a set of keys. In addition, the constraints can also be achieved by the key restrictions in the key space K, e.g., a temporary permission suspending according to key revocation mechanism. To sum up, the new cryptosystem based on this model can be naturally integrated into existing RBAC systems along with above-mentioned conditions.

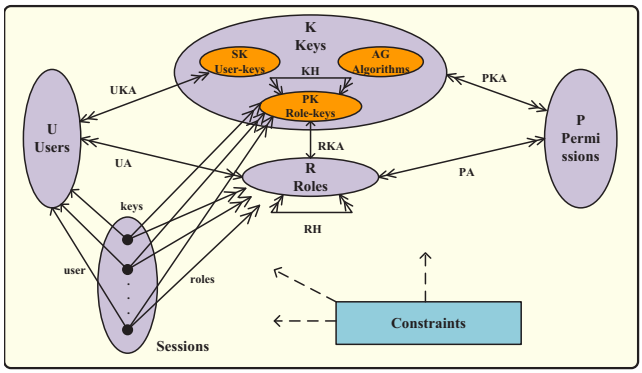


Fig. 2. A new RBAC model with role-key hierarchy

Our main objective is to map the role hierarchy in RBAC into a key management system. According to the condition 3 and 5, the role key set PK should have the same structure as the role hierarchy structure. Moreover, each user key $sk_{i,j} \in SK$ also needs to contain necessary information about role hierarchy for dealing with access functions independently by itself.

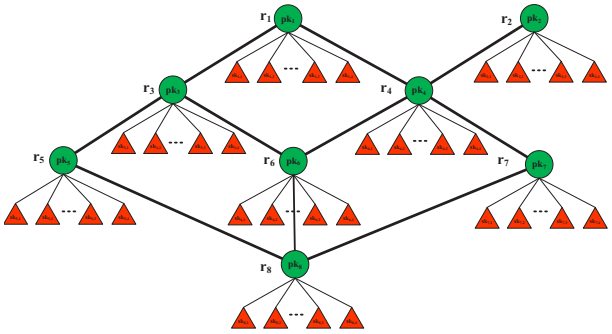


Fig. 3. Example of role-key hierarchy

In Fig. 3, we show an example of role-key hierarchy, in which a circle denotes a role key and a triangle denotes a user key, respectively. From this figure, we can easily observe two features:

- given a role r_i, all user keys sharing this role

$$\{sk_{i,j} : \forall u_{i,j}, (u_{i,j}, r_i) \in UA, (u_{i,j}, sk_{i,j}) \in UKA\}$$

correspond to a role key pk_i, where an unlimited number of users can belong to this role. This means that there is a one-to-many relationship between pk_i and $\{sk_{i,j}\}$ in a

cryptosystem built on this RKH. For example in Fig. 3, when the role key pk_6 is used to encrypt a message, all user keys $\{sk_{i,j}\}$ in this role can decrypt the message for $i = 6$.

- all role keys $\{pk_i\}$ constitute a key hierarchy KH, which has the same structure as RH in RBAC. The partial order relations in RH are still valid for $\{pk_i\}$ in KH. For the above example, when the message is encrypted by pk_6, all user keys $\{sk_{i,j}\}$ in r_1, r_2, r_3, r_4 can also decrypt the message for $i = 1, 2, 3, 4$.

To sum up, RKH provides a new cryptographic structure with one-to-many role/user key pair and partial-order relation of inheritance. This structure puts forward higher requirements for constructing the cryptosystems.

B. Role-based Cryptosystem

Given a role hierarchy $\Psi = \langle R, \preceq \rangle$ and a security parameter s, **Role-based Cryptosystem** (RBC) is a key management system that can construct a role-key hierarchy $\mathcal{H} = \langle U, K, R, P, \preceq \rangle$ on Ψ and generate all keys on $\mathcal{H}$, which is specified by three randomized algorithms, $Setup$, $KeyRGen$, and $AddUser$, described as follows:

- **Setup**(s, Ψ): Takes a security parameter s and a role hierarchy Ψ as an input. It produces a manager key mk and an initial parameter $params$, that is, $Setup(s, \Psi) \rightarrow \{\mathcal{H}, mk, params\}$.
- **GenRKey**$(params, r_i)$**:** Takes the parameter $params$ and a role index r_i. It generates a role key pk_i in r_i, that is, $KeyRGen(params, r_i) \rightarrow pk_i$.
- **AddUser**$(mk, ID, u_{i,j})$**:** Takes a user identity ID, a user index $u_{i,j}$, and the manager key mk. It outputs a user's secret key $sk_{i,j}$, which involves a user label $lab_{i,j}$ and a private key $dk_{i,j}$, for the user $u_{i,j}$, that is, $AddUser(mk, ID, u_{i,j}) \rightarrow sk_{i,j} = (lab_{i,j}, dk_{i,j})$. The user label $lab_{i,j}$ is added to the public encryption key: $params = params \bigcup \{lab_{i,j}\}$.

For ease of use, we expect that a RBAC system manager assigns the user key $sk_{i,j} = (lab_{i,j}, dk_{i,j})$ to a user, where $lab_{i,j}$ is a public label and $dk_{i,j}$ is a private key. This label $lab_{i,j}$ can be used to realize special functions such as designation, revocation, and tracing. Moreover, in public-key settings the permission process is performed only with the help of the public role key $\{pk_i\}$ containing the user's labels $\{lab_{i,j}\}$, which is also called as ID-based RBC because the user's public labels can be used to support the various functions.

C. Security Goal of RKH

Obviously, security requirements in general cryptosystem are not sufficient enough to reflect the requirements of role-key hierarchy. It is important to consider typical attacks when we try to design key hierarchy and its schemes. In contrast with existing key hierarchy, RKH has several unique features:

1) Each user $u_{i,j}$ is assigned to an exclusive user key $sk_{i,j}$, by which certain users can be chosen or identified in the processes of encryption, revocation, and tracing;
2) Public-key cryptography can be introduced to ensure the security of a user's private key even if the role key makes public in some systems. Therefore the role keys can be stored anywhere by RBAC systems; and
3) A tradition method for realizing partial order relation utilizes a derivation function on a user's private key, called $Delegate$, i.e., $Delegate(sk_{i,j}, r_l) = sk_{l,j}$ for $r_l \preceq r_i$, in most existing cryptosystems with partial-order property [8], [17]. However, this method is not conducive to various security mechanisms, such as key-based tracing and user-based revocation. Sometimes it might cause potential security vulnerabilities [18], [19], [20]. Therefore, the $Delegate$ function will be forbidden in our RBC system.

In order to ensure system security, RKH also needs to satisfy the following properties:

- Each user in a role cannot get permissions to access another role's objects except for its subordinates. Also, a user cannot forge other's secret keys;
- The role key can be modified to satisfy the requirements of constraint policy, but it should not interfere with the issued keys of others; and
- To support the capability of audit [21], there exists an efficient tracing algorithm to identify the corrupted users or gain the corresponding evidence.

The RKH-based system is, in essence, a group-oriented cryptography with "1:n" character, where one role key corresponds to many user keys. Hence, in addition to passive cryptanalysis, the collusion attack is a major attack, which focuses on changing the privilege of the granted users or getting the other users' keys. Let $\mathcal{R}_u$ denote the set of colluders. This kind of attack involves the following cases:

- Collusion attack for framing users, in which the corrupted users in $\mathcal{R}_u = \{u_{i_k,j_k}\}_{k=1}^{t}$ wish to forge a new or unused key in $U \setminus \mathcal{R}_u$ (called as honest user). The aim of this attack is to avoid tracing and frame innocent users.
- Collusion attack for role's privilege, in which the corrupted users in $\mathcal{R}_u = \{u_{i_k,j_k}\}_{k=1}^{t}$ wish to forge a new or unused key in $R \setminus R_{\mathcal{R}_u}$, where $R_{\mathcal{R}_u} = \{r \in R | u \in \mathcal{R}_u : (u, r) \in UA\}$. The aim of this attack is to change the privilege in partial order hierarchy.

We also present a formal security model for two cases of collusion attacks in Appendix 1, where the users are divided into two categories: honest users and corrupted users. The latter is used to build $\mathcal{R}_u$ [1]. The number of colluders $|\mathcal{R}_u| = t$ is an important parameter, where $|A|$ denotes the number of elements in the set A. Given a RBAC system with $|U| = n$ and $|R| = m$, a RBC scheme is to be (m, n, t)-collusion secure if the adversary cannot gain the advantage from $\mathcal{R}_u$ to break this scheme for any t-subset $\mathcal{R}_u \subseteq U$. It is said to be fully collusion secure when it is (m, n, n)-collusion secure.

IV. Security Mechanisms based on RKH

The role-based cryptosystem introduces a new key structure which is fully compatible with RBAC. Further, we are

[1] In real world $\mathcal{R}_u$ may be a set of all revoked users.

interested in building security mechanisms on this structure, especially for mechanisms which cannot be achieved by other cryptosystems. These security mechanisms are also specified by the algorithms in the algorithm set $AG \in K$. In this section, we provide such security mechanisms as follows:

A. Role-based Encryption

Encryption file systems allow users to encrypt resources (files or data) on disk, or synchronously transfer messages among multiple systems. Many encryption file systems have been developed in Windows and Linux environments, e.g., Windows Encrypting File System (EFS), SiRiUS [22] and Plutus [23]. However, these systems implement some trivial schemes where the number of ciphertexts in the file header grows linearly with the increased number of users who have permissions to access the file. To overcome such a limitation, we introduce a new scheme called **Role-based Encryption (RBE)**, which can be used to improve the performance of existing encryption file systems.

Definition 3 (Role-based Encryption): A role-based encryption scheme is an encryption system consisting of the following three procedures:

Initial: Takes role hierarchy $\langle R, \preceq \rangle$, and returns the role-key hierarchy $\mathcal{H} = \langle U, K, R, P, \preceq \rangle$ according to Setup and GenKey algorithms in RBC system;

Encrypt: Takes the encryption key pk_i and a plaintext M. It produces a ciphertext $\mathcal{C}_i$ on the role r_i: $Encrypt(pk_i, M, \mathcal{R}) \rightarrow \mathcal{C}_i$, where $\mathcal{R} = \mathcal{R}_r \cup \mathcal{R}_u$ is a set of revoked roles and users.

Decrypt: Takes the user key $sk_{i,j}$ and the ciphertext $\mathcal{C}_l$. It generates the plaintext M: $Decrypt(sk_{i,j}, \mathcal{C}_l) \rightarrow M$, where $r_l \preceq r_i$.

The relationship between encryption and decryption can be described as follows:

$$Decrypt(sk_{i,j}, Encrypt(pk_l, M, \mathcal{R})) = M$$

where $r_l \preceq r_i$, $(u_{i,j}, sk_{i,j}) \in UKA$, $(u_{i,j}, r_i) \in UA$, and the user is not a revoked user in $\mathcal{C}_l$, that is, $u_{i,j} \notin \mathcal{R}_u$ and $r_i \notin \mathcal{R}_r$. [2]

In order to improve the performance, we assume the following encrypted file structure: A file M is stored in the form $\langle Encrypt(pk_l, ek, \mathcal{R}), E_{ek}(M) \rangle$, where the former is called the file header, ek is a session key for encrypting M via a symmetric encryption method E, and $\mathcal{R}$ denotes the set of unauthorized roles and users. Such that, a user, who and the role of which are not in $\mathcal{R}$, can use his private keys $sk_{i,j}$ to decrypt the session key ek from $Encrypt(pk_l, ek, \mathcal{R})$ and then decrypt the file M from $E_{ek}(M)$. The cryptosystem based on this structure is also called as role-based encryption with revocation (called R-RBE, see Section VI) if $\mathcal{R} \neq \emptyset$.

Fig. 4 illustrates a role-based encryption file system constructed based on role-key hierarchy, where each role r_i is assigned to an encryption key pk_i and each user has a few decryption keys $\{sk_{i,j}\}$. An administrator only needs to keep the manager key mk, but the pk_i could be saved in the public directory of the system. When a user $u_{i,j}$ in r_i wants to create an encrypted file, the RBAC systems can automatically encrypt the file with a session key ek, then encrypt ek by using the user's current role (obtained from the current session s_i). The result is placed in the file header after the user gets the permissions from the RBAC systems. The user can also allow an arbitrary subset of authorized users to decrypt the file by performing proper assignments of unauthorized roles/users into $\mathcal{R}$ if necessary. When a user $u_{i,j}$ wants to access an encrypted file on r_l, the session key ek can be recovered by using $sk_{i,j}$ as long as the relation $r_l \preceq r_i$ holds and a revocation mechanism validates $r_i \notin \mathcal{R}_r$ and $u_{i,j} \notin \mathcal{R}_u$. Finally, the file is decrypted by ek after the user gets the access permissions from the RBAC systems.

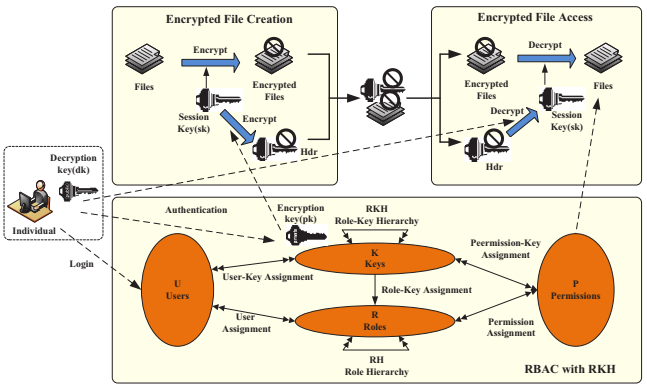


Fig. 4. Role-based encryption file system.

This scheme can provide the following security features for the encryption file systems:

1) Enabling better scalability because all users are organized into a uniformed role-based cryptographic framework. Most of cryptographic operations are performed at role level rather than at user level, e.g., the object of decryption is all users in a particular role and its seniors, rather than a single user or a role;
2) Protection against data leakage on the physical devices by using automatic encryption compatible with hierarchical RBAC, possibly caused by an untrusted administrator, a stolen laptop or a compromised server; and
3) Prevention of unauthorized data access by using a syncretic security mechanism based on dynamic cryptographic revocation technology.

B. Role-based Signature

The signature is a mathematical scheme for demonstrating the authenticity of a digital message or document. In RBAC model, the roles assigned to a user can be considered as one kind of identities of the user. Hence, a user could use his own roles to sign a resource. In other words, such a signature scheme provides a method to allow a user to *anonymously* sign a message on behalf of his roles. We call it **Role-based Signature (RBS)**. The formal definition of RBS is provided as follows:

[2] We also use the permission-to-key assignment relation to denote this relationship, that is, $(Decryption, (sk_{i,j}, pk_l, Decrypt)) \in PKA$.

Definition 4 (Role-based Signature): A role-based signature scheme is a digital signature consisted of the following four procedures:

Initial: Takes role hierarchy $\langle R, \preceq \rangle$, and returns the role-key hierarchy $\mathcal{H} = \langle U, K, R, P, \preceq \rangle$ according to Setup and GenRKey algorithms in RBC system;

Sign: Takes the role-key pk_i for r_i, a user key $sk_{i,j}$, and a message $M \in \{0,1\}^*$, and returns a signature σ: $Sign(pk_i, sk_{i,j}, M) \rightarrow \sigma$;

Verify: Takes the role-key pk_i and a signature σ on a message M. It returns the validation result which would be either valid or invalid. The latter response can mean either that σ is not a valid signature, or that the user who generated has been revoked (in a set of revoked users): $Verify(pk_i, \sigma, M) \rightarrow valid/invalid$;

Trace: Takes a user key $sk_{i,j}$ then this algorithm can trace a signature σ to at least one role member $u_{i,j}$ who generated it: $Trace(sk_{i,j}, \sigma) \rightarrow valid/invalid$.

The trace algorithm allows a third party to undo the signature anonymity using a special trapdoor and recognize the original signer. A secure role-based signature scheme must satisfy following properties:

- Correctness: This requires that, for all $K = (PK, SK)$ generated by role-key hierarchy, valid signatures by role members can always be verified correctly, and invalid signatures should fail in the verification phase:

$$Verify(pk_i, Sign(pk_i, sk_{i,j}, M), M) = valid.$$

- Unforgeability: Only members of a role can create valid signatures with the role.
- Anonymity: Given a message and its signature, the identity of the individual signer cannot be determined without the manager key mk.
- Revocation & Traceability: Given any valid signature, the manager or trusted third party (TTP) should be able to trace who issued the signature by the user's secret key. Given a revocation list $\mathcal{R}$, the revocation mechanism should be implemented if the manager or TTP checks whether or not the traced user in $\mathcal{R}$.
- Unlinkability: Given two messages and their signatures, we cannot determine whether the signatures were from the same signer or not.

In autonomous systems, role-based signature is used to verify the legality of the source of input data transmitted from other hosts or devices. This is more important for information sharing systems to prevent harmful information flows.

C. Role-based Authentication

Authentication allows access control systems to gain sufficient assurance that the identity of certain entity is legitimate as claimed. Cryptography-based authentication is widely adopted in current systems because it provides a higher level of security than password-based authentication. In addition, a real-time authentication for high-risk operations is necessary to prevent a user from changing roles after logging in. The authentication on RBAC should support two qualitative classes of identifications:

- **User-based authentication**, which is used to validate a user's identity, but the systems need to store the user's role information; and
- **Role-based authentication**, which can provide identifiable evidences that a given user possesses the attributes of a given role.

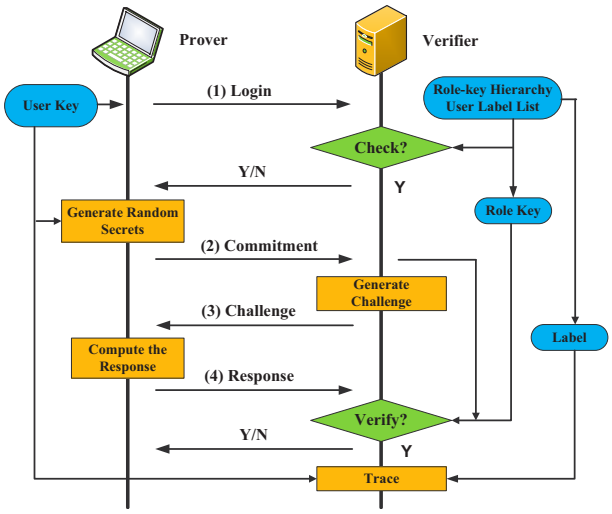


Fig. 5. Authentication protocol based on RBC.

Obviously, role-based authentication is a useful way for anonymous accesses, sharing systems, or off-line devices while the user information (including the user's public key in PKI) is not maintained by themselves. Furthermore, this approach can help achieve the interoperability as well. Hence, we propose a common framework of **Role-based Authentication (RBA)** based on a challenge-response protocol, as shown in Fig. 5, as follows:

Definition 5 (Role-based Authentication): A role-based authentication scheme is a challenge-response identification protocol between prover (P) and verifier (V), consisting of following four procedures:

Initial: Takes role hierarchy $\langle R, \preceq \rangle$, and returns the role-key hierarchy $\mathcal{H} = \langle U, K, R, P, \preceq \rangle$ according to $Setup$ and $GenRKey$ algorithms in RBC system;

Interact: the prover and the verifier execute the protocol:

1. **Login:** The prover sends the label of identity (including rolename and username) to the verifier, then the verifier checks the availability by searching user-label database or role hierarchy: $P \rightarrow V : r_i \vee lab_{i,j}$;
2. **Commitment:** If the check succeeds, the verifier requires the prover to return the commitment of the verifier's private key on a random number r: $P \rightarrow V : S = OneWay(sk_{i,j}, r)$;
3. **Challenge:** The verifier selects a challenge (random number) and sends it to the prover: $P \leftarrow V : c = Random()$;
4. **Response:** After receiving the challenge c, the prover computes the response in terms of his private key

$sk_{i,j}$ and the random numbers r in the commitment, and sends it back to the verifier: $P \rightarrow V : s = Respose(sk_{i,j}, r, c)$.

Verify: The verifier verifies whether the response is consistent with the commitment, the challenge, and the role key: $Verify(pk_i, S, c, s) \rightarrow valid/invalid$. In the case of user-based authentication, he can also check the validity of the prover's label: $Verify(pk_i, S, c, s, lab_{i,j}) \rightarrow valid/invalid$.

Trace: Takes a prover key $sk_{i,j}$ then it can analyze an existing record $re = \langle S, c, s, r_i \rangle$ to verify whether or not this prover generated this record: $Trace(sk_{i,j}, re) \rightarrow valid/invalid$.

Similarly to role-based signature, role-based authentication protocols must satisfy the following properties: correctness, anonymity, traceability, and unlinkability.

V. Proposed Schemes

In this section, we present our role-based cryptosystem scheme with role-key hierarchy based on pairing-based cryptosystem. Meanwhile, role-based signature & authentication and role-based encryption mechanisms are addressed based on the proposed role-based cryptosystem construction.

A. Bilinear Pairings

We set up our systems using bilinear pairings proposed by Boneh and Franklin [24], [25]. Let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ be three cyclic groups of large prime order p. $\mathbb{G}_1, \mathbb{G}_2$ are two additive group and $\mathbb{G}_T$ is a multiplicative group using elliptic curve conventions. Let $\hat{e}$ be a computable bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$[3] 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.

Where, $[a]P$ denotes the multiplication of a point P in elliptic curve by a scalar $a \in \mathbb{Z}_p$. A bilinear map group system $\mathbb{S}$ is a tuple $\mathbb{S} = \langle p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e \rangle$ composed of the objects as described above. $\mathbb{S}$ may also include group generators in its description.

B. Scheme for Role-based Cryptosystem

Let $\mathcal{H} = \langle U, K, R, P, \preceq \rangle$ is a role-key hierarchy with partial-order $\preceq$. Without loss of generality, we assume that the total number of roles and users are m and n in $\mathcal{H}$ respectively, i.e., $R = \{r_1, r_2, \cdots, r_m\}$, $|R| = m$, and $|U| = n$. We construct a RBC scheme as follows:

- $Setup(s, \Psi)$: 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 $H \in \mathbb{G}_2$, where $\mathbb{G}_1$ and $\mathbb{G}_2$ be bilinear group of prime order p. This algorithm first picks a random integer $\tau_i \in \mathbb{Z}_p^*$ for each role r_i in role-key hierarchy graph.[4] We define

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

Each τ_i is called as the secret of a role and U_i is the identity of a role. Further, it defines $U_0 = [\tau_0]G$ by using a random $\tau_0 \in \mathbb{Z}_p^*$. Thus, public parameter is

$$params = \langle H, V, U_0, U_1, \cdots, U_c \rangle$$

and we keep $mk = \langle G, \tau_0, \tau_1, \cdots, \tau_m \rangle$ secret.

- $GenRKey(params, r_i)$: This is an assignment algorithm for role encryption key from the setup parameter $params$. For a role r_i, the role key pk_i can be computed as follows:

$$\begin{cases} pk_i &= \langle H, V, W_i, \{U_k\}_{r_k \in \uparrow r_i} \rangle \\ W_i &= U_0 + \sum_{r_i \not\preceq r_k} U_k, \end{cases}$$

where, $\{U_k\}_{r_k \in \uparrow r_i}$ is the set of all roles in $\uparrow r_i$, which denotes the control domain for the role r_i. It is clear that $W_i = \left[\tau_0 + \sum_{r_i \not\preceq r_k} \tau_k\right] G$. For sake of simplicity, let $\zeta_i = \tau_0 + \sum_{r_i \not\preceq r_k} \tau_k$, so that we have $W_i = [\zeta_i]G$.

- $AddUser(mk, ID, u_{i,j})$: Given $mk = \langle G, \{\tau_i\}_{i=0}^m \rangle$ 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 $dk_{i,j} = \langle A_{i,j}, B_{i,j} \rangle$ where

$$\begin{cases} lab_{i,j} &= x_{i,j} &\in \mathbb{Z}_p^* \\ A_{i,j} &= \left[\frac{x_{i,j}}{\zeta_i + x_{i,j}}\right] G &\in \mathbb{G}_1, \\ B_{i,j} &= \left[\frac{1}{\zeta_i + x_{i,j}}\right] H &\in \mathbb{G}_2. \end{cases}$$

Finally, the above process outputs the set of role keys $\{pk_i\}$ and the set of user keys $\{sk_{i,j}\}$. More importantly, the security of user keys is not compromised even though all role keys are available in public. Furthermore, the total number of users is unlimited in each role.

C. Security Analysis of RBC Scheme

First, let us now turn to the problem of validity. We know that two arbitrary roles have one of three relations: $r_i \preceq r_j$, $r_j \preceq r_i$, and $r_i \parallel r_j$, so that partial order relation in role keys can be defined as

$$\sum_{r_i \not\preceq r_k} U_k = \sum_{r_k \in Ind(r_i)} U_k + \sum_{r_k \in Succ(r_i)} U_k,$$

where, $Ind(r_i) = \{r \in R | \ r \parallel r_i\}$ and $Succ(r_i) = \{r \in R | \ r \prec r_i\}$ denote the set of incomparable roles and successors for r_i, respectively.

This is illustrated in Fig. 6 (the top is senior-most roles and the bottom is junior-most roles) with the expression of W_i on the left of the node and U_i on the right. It is easy to find the following properties:

1) if $r_i \preceq r_j$, the representation of W_j (i.e., $\cup_{r_j \not\preceq r_k} \{U_k\}$ or $\cup_{r_j \not\preceq r_k} \{\tau_k\}$) contains that of W_i, e.g., $\{U_0\}_{W_8} \subseteq$

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

[4]Since the total number of roles is far less than the size of space of keys, we can use an efficient method to avoid the collision of value of role keys, e.g., the fast sort algorithm can be use to search the collision.

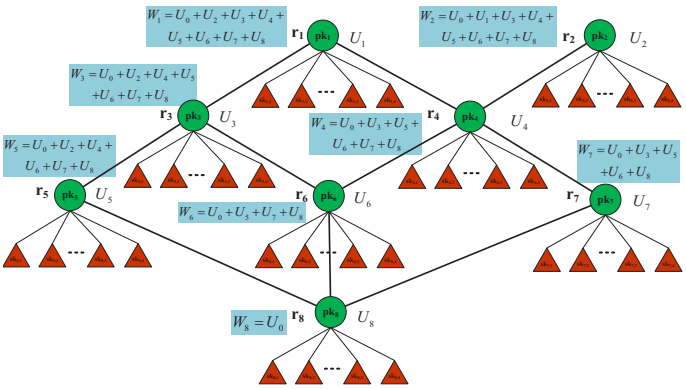


Fig. 6. Example of role-key relationship on RBC.

$\{U_0, U_5, U_7, U_8\}_{W_6}$ and $r_8 \preceq r_6$. So that a senior's W_j has much elements than a junior's W_i;

2) the representation of W_i is unique so that there do not exist W_i and W_j with the same representation, where $r_i \neq r_j$;
3) all representations of $\{W_i\}_{r_i \in R}$ contain all information in the corresponding RH, which means RH is hidden into KH (built on $\{W_i\}_{r_i \in R}$).

In addition, our scheme supports multiple top-most roles and multiple bottom-most roles. Moreover, for a bottom-most role, the value of W_i is not equal to 0, e.g., $W_8 = U_0$.

Next, we will make use of Theorems 1, 2, and 3 to prove the above-mentioned three properties, respectively. We first prove that this assignment works as required:

Theorem 1: Under the above assignment of $\{W_i\}_{r_i \in R}$, $(\cup_{r_j \not\preceq r_k}\{\tau_k\})_{W_j} \subset (\cup_{r_i \not\preceq r_k}\{\tau_k\})_{W_i}$ (or $(\cup_{r_j \not\preceq r_k}\{U_k\})_{W_j} \subset (\cup_{r_i \not\preceq r_k}\{U_k\})_{W_i}$) if and only if $r_j \prec r_i$. $(\cup_{r_j \not\preceq r_k}\{\tau_k\})_{W_j} = (\cup_{r_i \not\preceq r_k}\{\tau_k\})_{W_i}$ if and only if $r_j = r_i$.

The proof of this theorem is presented in Appendix 2. This theorem shows that the property 1) is correct, and the latter half of theorem also proves that the property 2) holds. More exactly, we will consider the value of W_i rather than the representation of W_i for the unique feature. Usually, we call *collision* if two random values are equal, i.e., $W_i = W_j$. Due to the reason that U_i is chosen at random, this scheme do not permit the collision among the role keys (or $W_i = W_j$), i.e., $pk_i = pk_j$ for $i \neq j$. The following theorem tells us that this collision probability is neglectable only if the security parameter κ is large enough. Moreover, the fast sort algorithm can help us to find the collision.

Theorem 2: The collision probability of getting any sum among m random integers, which are chosen in $\mathbb{Z}_p^*$ from a uniform distribution, is less than $\frac{(m+1)^2}{4p}$, where p is a large prime number.

The proof of this theorem is presented in Appendix 3. Since the total number of roles is far less than the size of space of keys, this theorem means that the collision probability is neglectable for $m \ll p$, e.g., given $m = 1000$ and $p \approx 2^k appa = 2^{160}$, the collision probability is less than $\frac{2^{20}}{2^{162}} = 2^{-142}$. Note that the security of RKH is not related to the combination of the role-keys, but rely heavily on the hardness of strong Diffie-Hellman (SDH) problem under the bilinear map group system (see Theorem 4).

Next, the following theorem indicates that the role hierarchy in $RBAC$ is hidden into role-key hierarchy (described as the property 3):

Theorem 3: Under the above assignment, role hierarchy is in one-to-one correspondence with key hierarchy.

A proof sketch for this theorem is presented in Appendix 4. According to this theorem, given all representations of $\{W_i\}_{r_i \in R}$, we show that role hierarchy may also be recovered by the following algorithm:

1) For each role r_i, it gets a set of roles $\cup_{r_i \not\preceq r_k}\{r_k\}$ from $\cup_{r_i \not\preceq r_k} U_k$ (the common element U_0 is excluded in this algorithm), then extracts its complementary set as $R_i = \cup_{r_i \preceq r_k}\{r_k\} = R \setminus \cup_{r_i \not\preceq r_k}\{r_k\}$. Further, it inserts each R_i into a record in the search table T.
2) While T is not empty, it does the following steps:
   a) Finds all records, which include only one element, sets these elements into the set of current roles C, then deletes these records from the table T;
   b) For each record $R_k \in T$, if $R_k \backslash C = \{r_i\}$, then outputs $r_i \prec_d r_j$ for all $r_j \in R_k \cap C$, else erases the elements in C, i.e., $R_k = R_k \setminus C$;

Based on this algorithm, Fig. 8 describes an example of extracting role hierarchy from key hierarchy in Fig. 6. We make use of $\rightarrow$ to denote $\succ_d$ in this figure.

First of all, the search table T is constituted according to all representations of $\{W_i\}_{r_i \in R}$, showed as the input list in Fig. 6. Secondly, the initial statements of T, showed in Iteration 1, are generated by running Step 1) in this algorithm. From these statements, we can find two records, R_1 and R_2, which include only one element, r_1 and r_2, respectively. Then, by running Step 2.b), the relations $r_1 \succ_d r_3$, $r_1 \succ_d r_4$, and $r_2 \succ_d r_4$ are outputted and the new statements of T are updated. Thirdly, we repeat above process to output relations $r_3 \succ_d r_5$, $r_3 \succ_d r_6$, $r_4 \succ_d r_6$, and $r_4 \succ_d r_7$ in Iteration 2. Next, we output results $r_5 \succ_d r_8$, $r_6 \succ_d r_8$, and $r_7 \succ_d r_8$ in Iteration 3. Finally, the bottom-most role r_8 is found in the last iteration, and then the algorithm halts. The outputted results are completely consistent with the original hierarchy.

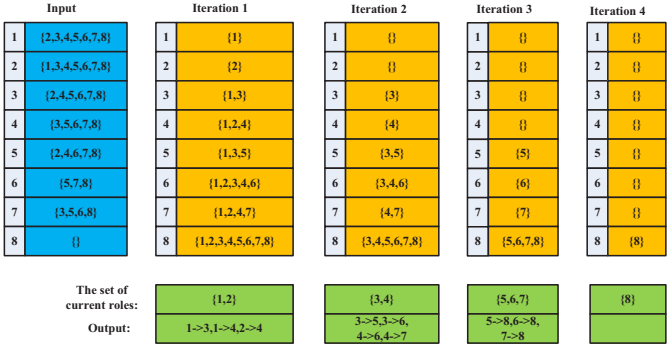


Fig. 8. Example of extracting role hierarchy from key hierarchy based on our algorithm.

$$
\begin{aligned}
V_{i,j}^{t} &= e\left(C_1 + \sum_{r_l \in \Gamma(r_j, r_i)} U_l', B_{j,k}\right) \cdot e\left(A_{j,k}, C_2\right) = e\left([t]W_i + \sum_{r_l \in \Gamma(r_j, r_i)} U_l', B_{j,k}\right) \cdot e\left(A_{j,k}, [t]H\right) \\
&= e\left([t]\left(U_0 + \sum_{r_i \not\preceq r_k} U_k\right) + \sum_{r_l \in \cup_{r_j \not\preceq r_k}\{r_k\} \setminus \cup_{r_i \not\preceq r_k}\{r_k\}} U_l', B_{j,k}\right) \cdot e\left(A_{j,k}, [t]H\right) \qquad (1) \\
&= e\left(\left[\zeta_j \cdot t\right]G, \left[\frac{1}{\zeta_j + x_{i,j}}\right]H\right) \cdot e\left(\left[\frac{x_{i,j}}{\zeta_j + x_{i,j}}\right]G, [t]H\right) = e(G,H)^{\frac{\zeta_j \cdot t}{\zeta_j + x_{j,k}}} \cdot e(G,H)^{\frac{t \cdot x_{i,j}}{\zeta_j + x_{j,k}}} = e(G,H)^{t} = V^{t}.
\end{aligned}
$$

Fig. 7. Equation for the validity of role-based encryption scheme.

We briefly analyze the performance of this algorithm. Let h is the height of role hierarchy. The algorithm recurs h times and the outputs of recurrence are all edges of one layer in role hierarchy. This means that this algorithm can recover the original role hierarchy in a polynomial running-time.

Finally, we show that our scheme is secure against collusion attack, in which two or more users, belonging to different roles, cooperate to discover a user key to which they are not entitled. This attack also called collusion attack with key hierarchy. To discuss the security against collusion, we make use of another hard problem, called strong Diffie-Hellman (SDH) problem, as follows:

Definition 6: [k-SDH problem]: Given $\langle G, [x]G, [x^2]G, \cdots, [x^k]G\rangle$ to compute $\langle c, \left[\frac{1}{x+c}\right]G\rangle$ where $c \in \mathbb{Z}_p^*$ and G be a generator chosen from $\mathbb{G}_1$ (or $\mathbb{G}_2$).

The standard collusion security is based on static colluders. Since we consider dynamic user management [5], we extend the security definition to the one that is more general than in [11]. That is, we allow the adversary to see the role keys before choosing the attacked users. Based on the collusion attacks in Section III-C, we have the following theorem:

Theorem 4: Given a role-key hierarchy $\mathcal{H} = \langle U, K, R, P, \preceq\rangle$, the role-based cryptosystem (RBC) scheme is (m, n, n)-collusion secure against collusion under Strong Diffie-Hellman (SDH) assumption.

We present a proof of this theorem in Appendix 5, where the whole proof for framing attack is given and the proof for role's privilege attack is stated briefly because it can be obtained from the former. The proof of this theorem indicates that the security is held even if G makes public. Moreover, this theorem clarifies that the security of this scheme is independent of the number of colluders, t. When $t = 1$, the proof of this theorem indicates that the security of the scheme against passive adversary (without collusion) is based on the hard problem $(G, [x]G) \rightarrow (c, [\frac{1}{x+c}]G)$ for $c \in_R \mathbb{Z}_p^*$.

D. Role-based Encryption Scheme

We adopt the RBC framework to build a lightweight role-based encryption (RBE) scheme, as follows:

- $Encrypt(pk_i, M)$: To encrypt the message $M \in \{0,1\}^*$, given any $pk_i = \langle H, V, W_i, \{U_k\}_{r_k \in \uparrow r_i}\rangle$ and an empty set of revoked users $\mathcal{R} = \emptyset$, the algorithm randomly picks $t \in \mathbb{Z}_p^*$ and then computes

$$
\begin{cases}
C_1 &= [t]\,W_i & \in \mathbb{G}_1 \\
C_2 &= [t]\,H & \in \mathbb{G}_2 \\
C_3 &= M \cdot V^t & \in \mathbb{G}_T \\
U_k' &= [t]\,U_k \in \mathbb{G}_1 & r_k \in \uparrow r_i
\end{cases}.
$$

  Finally, it outputs $\mathcal{C}_i = \langle C_1, C_2, C_3, \{U_k'\}_{r_k \in \uparrow r_i}\rangle$.
- $Decrypt(sk_{j,k}, \mathcal{C}_i)$: Given a ciphertext $\mathcal{C}_i$ from the role r_i, the k-th user in the role r_j can utilize the following equation to recover M from C_i with $dk_{j,k} = \langle A_{j,k}, B_{j,k}\rangle$, where $r_i \preceq r_j$:

$$
V_{i,j}^{t} = e\left(C_1 + \sum_{r_l \in \Gamma(r_j, r_i)} U_l', B_{j,k}\right) \cdot e\left(A_{j,k}, C_2\right),
$$

  where $\Gamma(r_j, r_i)$ denotes $\cup_{r_j \not\preceq r_k}\{r_k\} \setminus \cup_{r_i \not\preceq r_k}\{r_k\}$ and $r_l \in \Gamma(r_j, r_i) \subseteq\uparrow r_i$ in terms of Theorem 1. The algorithm outputs the session key $M = C_3 / V_{i,j}^t$.

The validity of this algorithm is guaranteed by Equation (1) in Fig. 7. Given a fixed role-key hierarchy, this algorithm achieves the constant length of ciphertexts and the optimal length of the user's secret keys $dk_{i,j}$, where the hidden constant relates to a couple of elements in a pairing-friendly group.

E. Role-based Signature & Authentication Scheme

The above construction can be applied to derive a role-based authentication (RBA) and signature (RBS) scheme. We propose a lightweight signature scheme to realize the anonymity and traceability. Further, this scheme can easily turn into a zero-knowledge RBA scheme. Given $\mathcal{H} = \{U, K, R, \preceq\}$, a user carries out the following process to sign a message M:

- $Sign(pk_i, sk_{i,j}, M)$: The signing algorithm takes a group public key $pk_i = (H, W_i, \{U_k\}_{r_k \in \uparrow r_i})$, a user private key $sk_{i,j} = (lab_{i,j}, A_{i,j}, B_{i,j})$, and a message $M \in \{0,1\}^*$, and proceeds as follows:
  1) Picks a random nonce $\alpha, \beta \leftarrow \mathbb{Z}_p^*$ and computes

$$
\begin{cases}
C_1 &= A_{i,j} + [\alpha]W_i \\
C_2 &= B_{i,j} + [\beta]H
\end{cases};
$$

  2) Picks blinding values $r \leftarrow \mathbb{Z}_p^*$ and compute helper values S:

$$
S = e(W_i, H)^r;
$$

  3) Computes a challenge value $c \in \mathbb{Z}_p^*$ using $Hash$:

$$
c = Hash(pk, M, C_1, C_2, S);
$$

[5]The users can be added and revoked into the system at any time.

$$e(W_i, B_{i,j}) \cdot e(A_{i,j}, H) = e\left([\zeta_j]\, G, \left[\frac{1}{\zeta_j + x_{i,j}}\right] H\right) \cdot e\left(\left[\frac{x_{i,j}}{\zeta_j + x_{i,j}}\right] G, H\right) = e(G, H) \quad (2)$$

$$\begin{aligned} R' &= \left(\frac{e\,(W_i, C_2) \cdot e\,(C_1, H)}{e(G, H)}\right)^{-c} \cdot e(W_i, H)^s = \left(\frac{e(W_i, B_{i,j} + [\beta]H) \cdot e(A_{i,j} + [\alpha]W_i, H)}{e(G, H)}\right)^{-c} \cdot e(W_i, H)^s \\ &= \left(\frac{e(W_i, B_{i,j}) \cdot e(W_i, [\beta]H) \cdot e(A_{i,j}, H) \cdot e([\alpha]W_i, H)}{e(G, H)}\right)^{-c} \cdot e(W_i, H)^s \\ &= (e(W_i, [\beta]H) \cdot e([\alpha]W_i, H))^{-c} \cdot e(W_i, H)^s = e(W_i, H)^{-c(\alpha+\beta)} \cdot e(W_i, H)^s = e(W_i, H)^r \end{aligned} \quad (3)$$

Fig. 9. Equation for the validity of role-based signature scheme.

4) Computes $s = r + c(\alpha + \beta)$:

Finally, the signature is $\sigma \leftarrow (C_1, C_2, c, s)$.

- $Verify(pk_i, \sigma, M)$: The verification algorithm takes a role key pk_i, a purported signature $\sigma = (C_1, C_2, c, s)$, and a message $M \in \{0,1\}^*$, and proceeds the following three steps:

1) Re-derives S as:

$$S' = \left(\frac{e\,(W_i, C_2) \cdot e\,(C_1, H)}{V}\right)^{-c} \cdot e(W_i, H)^s; \quad (4)$$

2) Computes $c' = Hash(pk, M, C_1, C_2, S')$;
3) Checks that the challenge c is correct if and only if $c = c'$. If matched, accept and reject otherwise.

The correctness of the verification procedure is guaranteed by using Equation (2) and (3) in Fig. 9.

- $Trace(\mathcal{R}_u, \sigma)$: The tracing algorithm takes a set of suspicious users $\mathcal{R}_u = \{(lab_{i,j}, B_{i,j})\}$ and a value $T = [\beta]W_i \in \sigma$ received from the $Sign$ algorithm. For each element $(lab_{i,j}, B_{i,j}) \in \mathcal{R}_u$, it checks whether $B_{i,j}$ is encoded in (T, C_2) by evaluating

$$e(W_i, C_2 - B_{i,j}) = e(T, H). \quad (5)$$

This user would be suspicious if this verification equation is accepted. [6]

The security of this scheme is guaranteed by the fact that $A_{i,j}$ and $B_{i,j}$ are kept private in C_1 and C_2. The distinct commitments, C_1, C_2 and S, could be generated to ensure the unlinkability by starting the algorithm with different random values α, β and r. In addition, the value W_i in Equation (4) and (5) determines that the user's role cannot be modified.

The (role-based and user-based) revocation mechanism can also be realized in our role-based signature scheme. Since the realization of role-based revocation is fairly simple and direct, we turn our attention to the realization of user-based revocation. We notice that the tracing algorithm can be used to implement the check of revoked users: given a set of revoked users $\mathcal{R}_u$, it first verifies that the signature σ is valid by using $Verify$ algorithm; then it ensures that σ is not generated by a revoked user (in $\mathcal{R}_u$) in terms of $Trace$ process. Therefore, when both conditions are held constant, the user's signature is accepted, otherwise, we say that it is a revoked signature. Here, we reiterate that the user's secret key $sk_{i,j}$ is secure even making $(lab_{i,j}, B_{i,j}) \in \mathcal{R}_u$ public. The reason is that the tracing or revocation check is based on $B_{i,j}$ in our scheme, as $A_{i,j}$ still keeps secret. Hence, the adversary cannot obtain the user's secret keys from a set of suspicious users.

[6]Given $sk_{i,j} = (lab_{i,j}, A_{i,j}, B_{i,j})$, the $Trace$ algorithm can be realized by using $S' \cdot (e(C_1 - A_{i,j}, H) \cdot e(W_i, C_2 - B_{i,j}))^c = e(W_i, H)^s$.

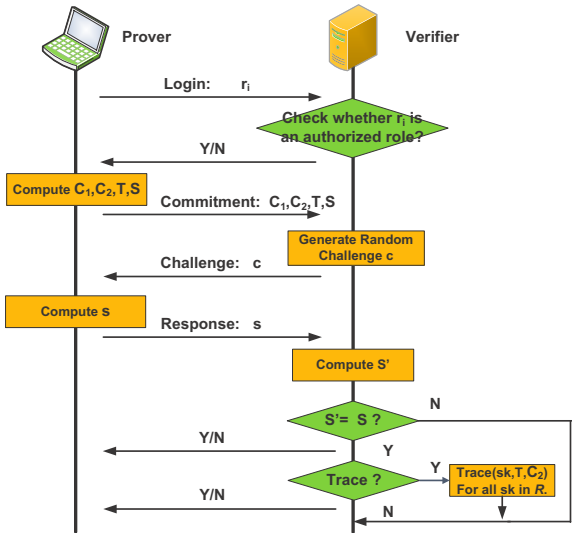


Fig. 10. Role-based authentication scheme.

The diagram of role-based authentication converted directly from our RBS scheme is shown in Fig. 10. This RBA scheme complies fully with 3-move Σ protocol (commitment, challenge and response), where the verifier's random challenge c instead of the hash value in RBS. After the interactive protocol, the verifier's verification rule is used to check whether $S' = S$ holds, where S' is computed by (4) and S is a commitment. The security of verification is guaranteed by a fact that it is infeasible to compute s according to the Discrete Logarithm assumption when C_1, C_2, T, S are fixed and c is a uniform random variable. Further, given a set of suspicious users, the Trace algorithm can be used for the revocation of user authorization. Note that, the leakage of $B_{i,j}$ in tracing process does not affect the security of user's private key because $A_{i,j}$ remains confidential.

VI. Implement of Revocation Mechanism

In this section, we propose a revocation mechanism [13] on the RBE scheme (called R-RBE scheme). This revocation mechanism can selectively change a set of authorized users and roles in one-time (or dynamic) encryption process.

A. RBE Scheme with Revocation Mechanism

Our revocation mechanism is able to revoke any number of users with no restriction of the revoked size. Let $\mathcal{R}$ denotes the

$$
\begin{aligned}
V_{i,j}^{t,\mathcal{R}_u} &= e\left([t]W_i + \sum_{r_l \in \Gamma(r_j,r_i)} U_l', B_{j,k}^{\mathcal{R}_u}\right) \cdot e\left(A_{j,k}, [t]B_{\mathcal{R}_u}\right) \\
&= e\left([\zeta_j \cdot t]\, G, \left[\frac{1}{\prod_{l=1}^{m}(\zeta_{i_l} + x'_{i_l,j_l}) \cdot (\zeta_j + x'_{j,k})}\right] H\right) \cdot e\left(\left[\frac{x'_{j,k}}{\zeta_j + x'_{j,k}}\right] G, \left[\frac{t}{\prod_{l=1}^{m}(\zeta_{i_l} + x'_{i_l,j_l})}\right] H\right) \\
&= e(G,H)^{\frac{\zeta_j \cdot t}{\prod_{l=1}^{m}(\zeta_{i_l} + x'_{i_l,j_l}) \cdot (\zeta_j + x'_{j,k})}} \cdot e(G,H)^{\frac{t \cdot x'_{j,k}}{\prod_{l=1}^{m}(\zeta_{i_l} + x'_{i_l,j_l}) \cdot (\zeta_j + x'_{j,k})}} = e(G,H)^{\frac{t}{\prod_{l=1}^{m}(\zeta_{i_l} + x'_{i_l,j_l})}} = (V_{\mathcal{R}_u})^t. \qquad (6)
\end{aligned}
$$

Fig. 11. Equation for the validity of role-based encryption scheme.

set of revoked roles and users. First, we modify the $AddUser$ algorithm in Section 5.2 as follows:

- $AddUser(mk, ID, r_i, u_{i,j})$: 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 secret value $x'_{i,j} = x_{i,j} - \sum_{r_i \not\preceq r_k} \tau_k \in \mathbb{Z}_p^*$. The user's public label is defined as

$$
\begin{cases}
lab_{i,j} &= \langle x_{i,j}, B_{i,j}, V_{i,j} \rangle, \\
B_{i,j} &= \left[\frac{1}{\zeta_i + x'_{i,j}}\right] H \quad \in \mathbb{G}_2, \\
V_{i,j} &= V^{\frac{1}{\zeta_i + x'_{i,j}}} \quad \in \mathbb{G}_T,
\end{cases}
$$

  The decryption key is defined as $dk_{i,j} = \langle A_{i,j} \rangle$, where $A_{i,j} = \left[\frac{x'_{i,j}}{\zeta_i + x'_{i,j}}\right] G \in \mathbb{G}_1$.

With help of the new public-label, the role-based encryption scheme with revocation mechanism is described as follows:

- $Encrypt(pk_i, M, \mathcal{R})$: Let $\mathcal{R} = \mathcal{R}_r \cup \mathcal{R}_u$ be a set of revoked roles and users, where $\mathcal{R}_u = \{u_{i_1,j_1}, \cdots, u_{i_m,j_m}\}$ and $\mathcal{R}_r = \{r_{k_1}, \cdots, r_{k'_m}\}$. To encrypt a random session key $ek \in \mathbb{G}_T$, given any $pk_i = \langle H, V, W_i, \{D_k\}_{r_k \in \uparrow r_i} \rangle$, the algorithm randomly picks $t \in \mathbb{Z}_p^*$ and then computes

$$
\begin{cases}
C_1 &= [t]\, W_i & \in & \mathbb{G}_1 \\
C_2 &= [t]\, B_{\mathcal{R}_u} & \in & \mathbb{G}_2 \\
C_3 &= M \cdot (V_{\mathcal{R}_u})^t & \in & \mathbb{G}_T \\
U'_k &= [t] D_k \in \mathbb{G}_1 & \exists & r_k \in \uparrow r_i
\end{cases}. \qquad (7)
$$

  where, $B_{\mathcal{R}_u}$ and $V_{\mathcal{R}_u}$ are efficiently computed from $\{B_{i_l,j_l}\}_{u_{i_l,j_l} \in \mathcal{R}_u}$ and $\{V_{i_l,j_l}\}_{u_{i_l,j_l} \in \mathcal{R}_u}$ by the aggregate algorithms (see section VI-C) as

$$
B_{\mathcal{R}_u} = \begin{cases} H, & \text{if} \quad \mathcal{R}_u = \emptyset, \\ \left[\frac{1}{\prod_{l=1}^{m}(\zeta_{i_l} + x'_{i_l,j_l})}\right] H, & \text{if} \quad \mathcal{R}_u \neq \emptyset, \end{cases}
$$

$$
V_{\mathcal{R}_u} = \begin{cases} V, & \text{if} \quad \mathcal{R}_u = \emptyset, \\ V^{\frac{1}{\prod_{l=1}^{m}(\zeta_{i_l} + x'_{i_l,j_l})}}, & \text{if} \quad \mathcal{R}_u \neq \emptyset. \end{cases}
$$

  The set $\{U'_k\}_{\exists r_k \in \uparrow r_i}$ may be a subset of $\{U'_k\}_{r_k \in \uparrow r_i}$ in order to revoke the roles in $\mathcal{R}_r$. Finally, it outputs the ciphertext $\mathcal{C}_i = \langle C_1, C_2, C_3, \{U'_k\}_{\exists r_k \in \uparrow r_i}, \mathcal{R} \rangle$.
- $Decrypt(sk_{j,k}, \mathcal{C}_i)$: 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 the private key $dk_{j,k} = A_{j,k}$ when $r_i \preceq r_j$ and $u_{j,k} \notin \mathcal{R}_u$:

$$
V_{i,j}^{t,\mathcal{R}_u} = e\left(C_1 + \sum_{r_l \in \Gamma(r_j,r_i)} U_l', B_{j,k}^{\mathcal{R}_u}\right) \cdot e\left(A_{j,k}, C_2\right), \quad (8)
$$

  where $\Gamma(r_j, r_i)$ denotes $\cup_{r_i \preceq r_l, r_j \not\preceq r_l} \{r_l\}$, $U'_l \in \{U'_k\}_{\exists r_k \in \uparrow r_i}$ for all $r_l \in \Gamma(r_j, r_i)$, and

$$
B_{j,k}^{\mathcal{R}_u} = \begin{cases} B_{j,k}, & \text{if} \quad \mathcal{R}_u = \emptyset, \\ \left[\frac{1}{\prod_{l=1}^{m}(\zeta_{i_l} + x'_{i_l,j_l}) \cdot (\zeta_j + x'_{j,k})}\right] H, & \text{if} \quad \mathcal{R}_u \neq \emptyset, \end{cases}
$$

  from $\{B_{i_l,j_l}\}_{u_{i_l,j_l} \in \mathcal{R}_u}$ and $B_{j,k}$. The algorithm produces the session key $ek = C_3 / V_{i,j}^{t,\mathcal{R}_u}$.

When $\mathcal{R}_r = \mathcal{R}_u = \emptyset$, the R-RBE scheme is reduced to the RBE scheme in Section V-D. Thus, the RBE scheme is a trivial construction of the R-RBE scheme.

B. Analysis of Correctness for Revocation

We analyze the validity of our scheme in two cases:

1) **Role-based revocation**. Given a set $\mathcal{R}_r$, our scheme supports role revocation by customizing the elements of $\{U'_k\}_{r_k \in \uparrow r_i}$ in (7), i.e., generating $\{U'_k\}_{\exists r_k \in \uparrow r_i}$ for a specified subset of roles in $\uparrow r_i$. We first establish a role revocation table, shown in Table I. Assume that the role key pk_i of r_i is used to encrypt message. The essential elements of $\{U'_k\}$, listed in this table, can be used to ensure that all private key $\{sk_{j,k}\}$ of r_j is able to decrypt the message. For example in Fig.6, when $r_i = r_6$ and $r_j = r_3$, the essential elements of $\{U'_k\}$ is three values U'_2, U'_4, U'_6, which ensure that all users in r_3 are authorized to decrypt the message.

At the same time, in this table we list "all authorized roles" dominated by these essential elements based on partial order relation. In the above-mentioned example, in terms of $r_6 \preceq r_3$, we know that the role r_6 is also the authorized roles dominated by the elements U'_2, U'_4, U'_6. So that all authorized roles of r_3 include r_3 and r_6. Inversely, this means that the set $U' = \{U'_2, U'_4, U'_6\}$ can be used to revoke the set of roles $\mathcal{R}_r = (\uparrow r_6) \backslash \{r_3, r_6\} = \{r_1, r_2, r_4\}$. However, we must note that, from this table, not any $\mathcal{R}_r$ can be revoked in $\uparrow r_i$ because of constraints of partial order relation.

2) **User-based revocation**. Given a set $\mathcal{R}_u$, user revocation is implemented by checking whether $B_{\mathcal{R}_u}$, $B_{j,k}^{\mathcal{R}_u}$, and $V_{\mathcal{R}_u}$ can be computed, as well as error checking for dividing by zero. We explain this kind of revocation by using a simple $\mathcal{R}_u$ with two users, as follows: with respect to the definition of $x_{i,j}$ and $x'_{i,j}$, we have

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

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 $\mathcal{R}_u = \{u_{i_l,j_l}, u_{i_k,j_k}\}$ and

TABLE I
EXAMPLE FOR ROLE REVOCATION TABLE.

r_i	r_j	The element of $\{U'_k\}$	All authorized roles
r_6	all	$U'_1, U'_2, U'_3, U'_4, U'_6$	r_1, r_2, r_3, r_4, r_6
	r_1	U'_2, U'_3, U'_4, U'_6	r_1, r_3, r_4, r_6
	r_2	U'_1, U'_3, U'_4, U'_6	r_2, r_4, r_6
	r_3	U'_2, U'_4, U'_6	r_3, r_6
	r_4	U'_3, U'_6	r_4, r_6
	r_6	$\emptyset$	r_6
r_7	r_1	D_2, D_4, D_7	r_1, r_4, r_7
	r_2	D_1, D_4, D_7	r_2, r_4, r_7
	r_4	D_7	r_4, r_7
	r_7	$\emptyset$	r_7

$i_l \neq i_k$, it is easy to obtain

$$\left[\frac{1}{x_{i_l,j_l}-x_{i_k,j_k}}\right](B_{i_k,j_k} - B_{i_l,j_l}) = \left[\frac{1}{(\zeta_{i_l}+x'_{i_l,j_l})(\zeta_{i_k}+x'_{i_k,j_k})}\right] H = B_{\mathcal{R}_u}, \quad (9)$$

and

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

Similarly, $B_{\mathcal{R}_u}$, $B^{\mathcal{R}_u}_{j,k}$, and $V_{\mathcal{R}_u}$ can be efficiently computed in an arbitrary revocation set $\mathcal{R}_u$ by a general recursive method, which is defined in Subsection VI-C. Therefore, we can prove (8) by (6). The revocation mechanism can be supported by (9). That is, for $u_{i,j} \in \mathcal{R}_u$, $B^{\mathcal{R}_u}_{j,k}$ cannot be computed because the denominator can be zero in a fraction $\frac{1}{x_{i,j}-x_{i',j'}}$, where $u_{i',j'} \in \mathcal{R}_u$. Note that, this kind of revocation is not subjected to the user's role, as well as the size of $\mathcal{R}_u$. Such that our scheme can be used to revoke the keys of an unlimited number of users.

Note that, the user revocation method can be used to permanently prevent the revoked users from accessing the encrypted resources if the values, $\mathcal{B}_{\mathcal{R}_u}$ and $\mathcal{V}_{\mathcal{R}_u}$, are published in the public parameters by replacing H and V for a public set of permanently revoked users $\mathcal{R}_u$.

C. Aggregate Algorithms for User Revocation

It is more important to compute three values $B^{\mathcal{R}_u}$, $V^{\mathcal{R}_u}$, and $B^{\mathcal{R}_u}_{j,k}$ from the user's labels in an efficient way. We provide such a recursive method (called as aggregate algorithm) to solve this question, as follows:

Given $\mathcal{R}_u = \{x'_{i_1,j_1}, \cdots, x'_{i_m,j_m}\}$ and their labels $\{lab_{i_k,j_k}\}$ for $k \in [1,m]$ 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 Equation (9), for all $k, l \in [1,m]$, it is easy to obtain the equation

$$B_{i_k,j_k} - B_{i_l,j_l} = \left[\frac{1}{\tau_0+x_{i_k,j_k}}\right] H - \left[\frac{1}{\tau_0+x_{i_l,j_l}}\right] H = \left[\frac{x_{i_l,j_l}-x_{i_k,j_k}}{(\zeta_{i_l}+x'_{i_l,j_l})(\zeta_{i_k}+x'_{i_k,j_k})}\right] H.$$

To expand this equation to multi-user cases, we define the following denotation $\widetilde{B}_{s,r}$ for any pair (s,r), where $1 \leq s < r \leq m$, $\widetilde{B}_{s,r} = \left[\frac{1}{\tau_0+x_{i_r,j_r}} \cdot \frac{1}{\prod_{k=1}^{s}(\tau_0+x_{i_k,j_k})}\right] H$.

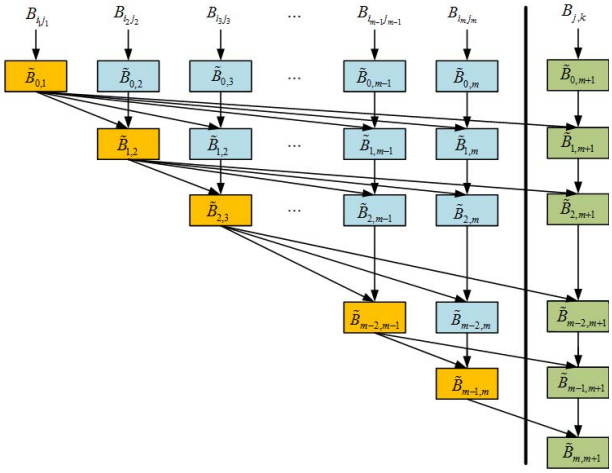


Fig. 12. The diagram flow of generation of $\widetilde{B}_{s,r}$ and $B^{\mathcal{R}_u}_{j,k}$.

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

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

where $\widetilde{B}_{0,r}$ is defined as the initial input B_{i_k,j_k} for $k = [1,r]$. Obviously, we can get $B^{\mathcal{R}_u}_{j,k}$ in the same way, or it can be computed from the resulting sequence $(B_{i,j}, \langle \widetilde{B}_{0,1}, \widetilde{B}_{1,2}, \cdots, \widetilde{B}_{m-1,m}\rangle)$, where

$$\begin{cases} \widetilde{B}_{0,m+1} = B_{j,k}, \\ \widetilde{B}_{s,m+1} = \left[\frac{1}{x'_{j,k}-x'_{i_s,j_s}}\right](\widetilde{B}_{s-1,s} - \widetilde{B}_{s-1,m+1}), \\ \qquad \forall s \in [1,m], \\ B^{\mathcal{R}_u}_{j,k} = \widetilde{B}_{m,m+1}. \end{cases}$$

Similarly, we defines $\widetilde{V}_{s,r} = V^{\frac{1}{\tau_0+x_{i_r,j_r}} \cdot \frac{1}{\prod_{k=1}^{s}(\tau_0+x_{i_k,j_k})}}$, and computes $V_{\mathcal{R}_u}$ from $V_{i_1,j_1}, \cdots, V_{i_m,j_m}$ as follows:

$$(V_{\mathcal{R}_u} = \widetilde{V}_{m-1,m}) \leftarrow \begin{cases} \widetilde{V}_{0,r} = V_{i_r,j_r} \quad \forall r \in [1,m] \\ \widetilde{V}_{s,r} = \left(\frac{\widetilde{V}_{s-1,s}}{\widetilde{V}_{s-1,r}}\right)^{\frac{1}{x_{i_r,j_r}-x_{i_s,j_s}}} \\ \qquad \forall s \in [1,m-1], \forall r \in [s+1,m] \end{cases}.$$

D. Security Analysis

As an encryption scheme, a primary requirement for RBE and R-RBE is to provide semantic security. Also, collusion attack must be taken into account in investigating semantic security as a group-oriented cryptosystem. This kind of collusion is most likely to occur between an adversary and some internal traitors, who hold some valid keys [7]. Hence, we define a semantic security against adaptively chosen plaintext attack with dynamic colluders (called IND-dcCPA). This security is

[7]Usually, the internal traitors are considered as the revoked users in the analysis. The security of role-based revocation is easy-to-understand, so that we will focus on the analysis of user-based revocation (or collusion attack).

defined using the following game between an attack algorithm $\mathcal{A}$ and a challenger $\mathcal{B}$, where $\mathcal{A}$ corrupts some $u_{i,j} \in \mathcal{R}_u$ to decrypt $\mathcal{C}_k$, even if $u_{i,j} \in r_i$ and $r_i \preceq r_k$, for a set of colluders $\mathcal{R}_u$. This game is defined as follows:

1. Initial. $\mathcal{B}$ constructs an arbitrary $\mathcal{H}$, and then runs Setup algorithm and gives $\mathcal{A}$ the resulting parameters $params$ and $\mathcal{H}$, keeping mk secret.

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

- **(a) Public Label Query** ($u_{i,j} \notin \mathcal{R}_u$)**.** 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** ($u_{i,j} \in \mathcal{R}_u$)**.** 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_k on which it wishes to be challenged. $\mathcal{B}$ picks a random bit $b \in \{0, 1\}$ and sends the challenge ciphertext $\mathcal{C}_k = Encryt(\mathcal{R}_u, pk_k, M_b)$ to $\mathcal{A}$, where for $\forall u_{i,j} \in \mathcal{R}_u$, $u_{i,j} \in r_i$ and $r_i \preceq r_k$.

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 in $\mathcal{R}_u$, collude to try and expose a ciphertext intended for users in $U \backslash \mathcal{R}_u$ only. The set $\mathcal{R}_u$ is adaptively chosen by the adversary. In this game, we define the advantage of the adversary $\mathcal{A}$ in attacking the scheme as

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

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

We prove the semantic security of our R-RBE scheme under the assumption of GDDHE problem.

Definition 7 ((n,t)-GDDHE$_1$ Problem): Let $f(x)$ and $g(x)$ be two known random polynomials of respective degree t and $n - t$ with pairwise distinct roots, i.e.,

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

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

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

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 two generators of $\mathbb{G}_1$ and $\mathbb{G}_2$. For any algorithm $\mathcal{A}$ that makes a total of at most q queries to the oracles computing the group operation in $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ and the bilinear pairing e, the advantage of $\mathcal{A}$ is $Adv^{ind}_{\mathcal{E},\mathcal{A}}(n, t) \leq \frac{(q+2(n+t+4)+2)^2 \cdot (2n)}{2p}$.

The (n,t)-GDDHE$_1$ problem has been proved to be NP-hard in the generic bilinear groups in [26]. Next, we prove that our RBE scheme with revocation mechanism is semantically secure against the above-mentioned game in the following theorem:

Theorem 5: The (m, n, t)-RBE is semantically secure against dynamic colluders (IND-dcCPA) 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 $Adv^{ind}_{\mathcal{E},\mathcal{A}}(m, n, t) \leq \frac{(q+2(n+t+4)+2)^2 \cdot (2n)}{p}$.

We present a proof of this theorem in Appendix 6.

VII. Performance Evaluation

An experimental role-based cryptosystem was implemented to test the feasibility of our schemes. This system was developed with a standard C++ language in QT environment, which supports cross-platform deployment. This system consists of three modules: RBC module, access control module and application module. In RBC module, we adopted GNU multiple precision arithmetic library (GMP) to handle integers of arbitrary precision. Then, a finite fields arithmetic library was constructed to realize the run-time environment of elliptic curve and pairing-based cryptosystems. In addition, a cryptographic access control library was developed based on the finite fields arithmetic library to realize various proposed RBC algorithms. Finally, the RBE/RBA/RBS algorithms worked with a lightweight access control module to provide encryption, authentication and key-label management services for the application module.

Scalability. The experimental results show our constructions are able to provide better scalability, which is an important requirement for RBAC [16]. The notion of scalability is multi-dimensional. In our schemes we can achieve scalability with respect to the number of roles, the size of role hierarchy, cardinality on user-role assignments, and so on. In our scheme the scalability is exhibited in the following aspects:

- Supports an unlimited number of users, where a new user can join anytime without change of pre-existing user keys nor ciphertext size;
- Supports a large-size of role hierarchy with arbitrary complex structures, where the size of each user's private key is fully independent from the number of roles and role hierarchy;
- Supports an effective approach to revoke (dynamically or permanently) any subgroup of users;
- Provides a good tracing ability owning to the uniqueness of user's private key, where the tracing overheads are directly proportional to the number of suspicious users;
- Provides collusion-secure for arbitrarily large collusions of users, as well as role-based off-line authentication without any data exchange of user information between the verifier and the system manager.

Table II summarizes existing group-oriented cryptosystems currently available. Here dynamic and static revocation are

TABLE II
COMPARISONS OF SCHEMES (n IS THE NUMBER OF USERS AND m IS THE NUMBER OF ROLES OR GROUPS).

	System Type	Commu. Overhead	Private Key Size	Public Key Size	Comput. Cost	Partial-Order Relation	Revocation	Signature	Delegation
[5]	SKM[‡]	$O(n^{1/k})^{\dagger}$	$O(\log n)$	None	$O(\log n)$	Tree Structure	None	N/A	None
[27]	SKM[‡]	$O(\log^2 n)$	$O(\log n)$	None	$O(\log n)$	Tree Structure	None	N/A	None
[11]	BE[§]	$O(1)$	$O(1)$	$O(n)$	$O(n)$	None	None	None	None
[28]	BE[§]	$O(\sqrt{n})$	$O(\sqrt{n})$	$O(\sqrt{n})$	$O(\sqrt{n})$	None	Yes	None	None
[15]	HKM[♮]	N/A	$O(1)$	$O(m^2)$	$O(1)$	General Structure	None	N/A	Yes
[14]	TTP-RBE[♯]	$O(1)$	$O(1)$	$O(n+m)$	$O(n+m)$	General Structure	Static	None	None
[9]	GIB-BE[◇]	$O(1)$	$O(m)$	$O(m)$	$O(m)$	General Structure	None	None	Yes
Our Scheme	RBC	$O(m)$	$O(1)$	$O(m)$	$O(m)$	General Structure	Dynamic/Static	Yes	None

[†] k is the number of stratified subsets to obtain a reasonable computation cost, i.e., when n is less than one trillion, $n^{1/8} < \log n$. [‡] SKM: Symmetric-key Key Management (logical key hierarchy (LKH) scheme and subset difference (SD) scheme); [♮] HKM: Hierarchical Key Management; [§] BE: public-key Broadcast Encryption; [♯] TTP-RBE: RBE with Trusted Third Party (TTP); and [◇] GIB-BE: Generalized Identity Based and Broadcast Encryption.

used to denote temporary and permanent revocation of users, respectively. It is easy to find that our scheme has better performance and efficiency on security mechanisms. Therefore, our cryptosystem can be applied to large-scale role-based information systems, such as healthcare and financial systems.

Computational Cost. The basic operation of our scheme is to compute a multiple of a point in an elliptic curve, namely, $[k]P$, where k is a positive integer and P is an elliptic curve point. We neglect the costs of an addition of two points and a modular arithmetic operation because they run fast enough. Another important operation is to compute of a bilinear map $e(\cdot,\cdot)$ between two points. Therefore, we use the costs of multiple operation and bilinear map operation to measure the computation complexity of our schemes. In Table III the costs of various algorithms in RBE, RBS, and RBA schemes are listed, where / is a separator character, and n/m denotes the number of multiple operations and bilinear map operations, respectively. In Table II we also summarize the computational costs of different schemes. From these two tables, it is easy to find that the costs of our RBE scheme are proportional to the size m of roles regardless of the size n of users. The size of roles is relatively smaller than that of users in a large-scale system, therefore our scheme can potentially perform better.

TABLE III
COMPARISON OF COMPUTATION COSTS ON RBC.

	RBE	RBS/RBA
Setup	$(m+1)$/1	$(m+1)$/1
GetRkey	0/0	0/0
AddUser	2/0	2/0
Encrypt/Sign	$(m+3)$/0	4/1
Decrypt/Verify	1/2	5/3
/Trace	/	1/2

Let us now turn our attention to the aggregate algorithm (see Section VI-C) in our R-RBE scheme. Let $t = |\mathcal{R}_u|$ be the size of revoked users. In the encryption process, the overhead of aggregate algorithm is $t(t-1)/2 \cdot (t_p + t_{\mathbb{G}})$, where t_p is the running time of a subtraction and an inversion modulo $p = |\mathbb{G}|$ and $t_{\mathbb{G}}$ is the time of an exponentiation and a division in $\mathbb{G}$. Similarly, the overhead is $(t+1) \cdot (t_p + t_{\mathbb{G}})$ in the decryption process. In summary, It is quite clear that all our schemes, RBE, R-RBE, RBS, and RBA, have low computational costs.

Communication Overhead. With the same assumption of scalability, we estimate the influence of communication overloads (ciphertext size) under the different scales. Suppose the security parameter κ is 80-bits, we need the elliptic curve domain parameters over $\mathbb{F}_q$ with $|q| = 160$-bits [8]. This means that the length of integer is $l_0 = 2\kappa$ in $\mathbb{Z}_p$. Similarly, we have $l_1 = 4\kappa$ in $\mathbb{G}_1$, $l_2 = 20\kappa$ in $\mathbb{G}_2$, and $l_T = 10\kappa$ in $\mathbb{G}_T$ when the embedding degree is 5. For RBS/RBA scheme, the communication overloads of $Sign/Interact$ is $2l_0+2l_1+l_2 = 32\kappa = 320$ bytes. For RBE scheme, the length of ciphertext is at most $(m+1)l_1 + l_2 + l_T = 4m\kappa + 34\kappa = 340 + 40m$ bytes. In terms of the size of role hierarchy, we can easily compute that the overheads are increased from 0.7 KBytes (for 10 roles) to 40 KBytes (for 1000 roles). Further, for R-RBE scheme, the length of ciphertext is at most $(m+1)l_1+l_2+l_T+t\cdot l = 4m\kappa+34\kappa+16t = 340+40m+tl$ bytes, where l is the length of user's label. In contrast with RBE, the revocation mechanism only takes a few space of list of revoked users because our RBE scheme is a special case of the R-RBE scheme when $\mathcal{R} = \emptyset$. In Table II we also show the communication overhead of different schemes. Our scheme with worst-case overhead of $O(m)$ is not optimal because of existence of revocation mechanisms, but it is still effective for large-scale practical applications.

Experimental Comparison. 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 header 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 256KB 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." [29] We show such a structure in Fig. 13 (Top). In contrast to the above, the RBAC systems built on our RBE scheme (see Section IV-A) can automatically use the role key to encrypt the files in terms of the user's role r_i in a transparent way. Such a file header is also shown in Fig. 13

[8]Elliptic curve domain parameters over $\mathbb{F}_p$ with $\lceil \log_2 p \rceil = 2\kappa$ supply approximately κ bits of security, which means that solving the discrete logarithm problem on associated elliptic curve is believed to take approximately 2^{κ} operations.

(Bottom), in which "RBE Ciphertext" consists of the constant-size (C_1, C_2, C_3) in a ciphertext $\mathcal{C}_i$, "Authorized Role Table" consists of $\{U'_k\}_{\exists r_k \in \uparrow r_i} \in \mathcal{C}_i$, and "Revoked User Table" consists of all user's labels in $\mathcal{R}_u$. By using such a revocation mechanism, the number of users is not limited. Moreover, for a new user, no files need to be changed to permit the access of the existing files.

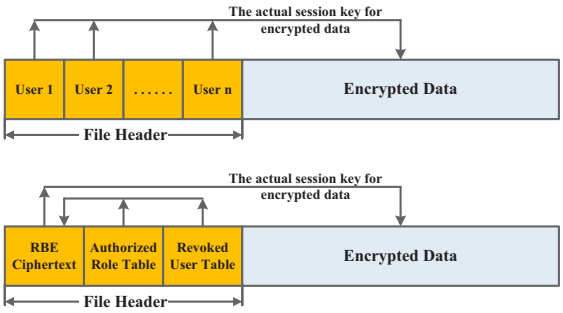


Fig. 13. Comparison between exiting encrypted file system in Windows NT (Up) and our scheme (Down).

For the sake of clarity, we evaluate the performance of EFS on our R-RBE scheme as follows: Suppose κ is 80-bits [30], [31] and an elliptic curve over $\mathbb{Z}_p$ with $|p| = 160$-bits. This means that the length of integer is $l_0 = 2\kappa$ in $\mathbb{Z}_p$ and the length of elements in $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ satisfies $l_1 = 4\kappa$, $l_2 = 20\kappa$, and $l_T = 10\kappa$, respectively. We assume that the embedding degree of elliptic curve is 5. In R-RBE, the length of the file header is $320m + 20t + 2,720$ bits, where l is the length of user's label being set to 20 bits presumed. Considering a system where each role contains 40 users on average, with 20 roles, 800 users and 100 revoked users, the file header is just $320 \times 20 + 20 \times 100 + 2,720 = 11,120\text{bits} \approx 1.36\text{KB}$.

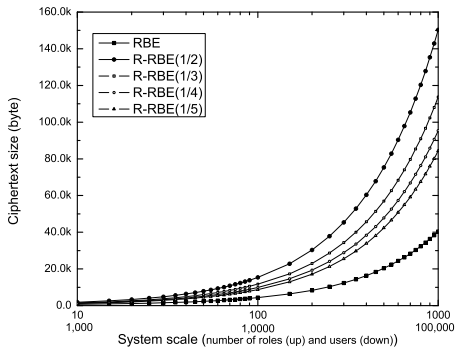


Fig. 14. The ciphertext size under different scales.

In contrast to the above EFS in Microsoft NT (256KB), this storage cost is far less than 256KB, which is the file header size in existing EFS. Furthermore, in EFS based on RBE, the file header with 256KB can support the system with about 2,000 roles and each role containing 300 users on average and 30,000 revoked users, where the length of the user label is 40bit. In theory, the above-mentioned system can support unlimited number of users, which is much more efficient than existing EFSs.

In Fig. 14, we show the change rate of ciphertext size for the RBE schemes, which includes two versions: basic RBE scheme and R-RBE scheme. In this figure, the size of roles changes from 10 to 1000, each role has 100 users, and the number of revoked users in each role is equal to $1/2$, $1/3$, $1/4$, and $1/5$ the number of roles. According to the parameters in the above analysis, Fig. 14 also indicates that the size of revoked users has more impact than other factors. Our results also indicate that the encryption based on RBE scheme performs far better than the conventional encryption file systems (EFS) with the following parameters:

- Even if we deal with a large-scale organization of 500,000 users the header of a file only requires 256 KBytes in theory (using a standard 10-bytes (80-bits) security parameter); and
- EFS with our scheme can revoke an approximate 1,000 users (some intermediate data are saved to decrypt a file faster) or 10,000 users in a compressed form at once.

VIII. Conclusion

We have proposed a role-key hierarchy structure along with hierarchical RBAC model to accommodate the requirements of cryptographic access control for large-scale systems. Based on this hierarchy model, we further proposed several practical role-based security mechanisms to support signature, authentication and encryption constructions on elliptic curve cryptosystem. Our experiments clearly demonstrated the proposed schemes are flexible and efficient enough to support large-scale systems. For our further work, we plan to accommodate other access control features of RBAC such as session management and constraints. Also, our promising results lead us to investigate how emerging distributed computing technologies such as service computing, cloud computing and mobile computing can leverage the proposed schemes with possible extensions.

Acknowledgment

The authors are indebted to anonymous reviewers for their valuable suggestions. The authors thank Zexing Hu and Huaixi Wang, two students at Peking University, for discussing and verifying the scheme using the C++ language.

References

[1] R. Sandhu, D. Ferraiolo, and D. Kuhn, "The nist model for role-based access control: Towards a unified standard," in *Proceedings of 5th ACM Workshop on Role Based Access Control (RBAC'00)*, 2000, pp. 47–63.

[2] J. Jing and G.-J. Ahn, "Role-based access management for ad-hoc collaborative sharing," in *Proc. of 11th Symposium on Access Control Models and Technologies (SACMAT)*, 2006, pp. 200–209.

[3] S. Akl and P. Taylor, "Cryptographic solution to a multilevel security problem," in *Advances in Cryptology (CRYPTO'82)*, 1982, pp. 237–249.

[4] ——, "Cryptographic solution to a problem of access control in a hierarchy," *ACM Transaction Computer System*, vol. 1, no. 3, pp. 239–248, 1983.

[5] D. Wallner, E. Harder, and R. Agee, "Key management for multicast: Issues and architecture," In internet draft draft-waller-key-arch-01.txt, Tech. Rep. IETF RFC 2627, 1999.

[6] C. Wong, M. Gouda, and S. Lam, "Secure group communications using key graphs," in *Proc. ACM SIGCOMM'1998*, vol. 28 of ACM press, 1998, pp. 68–79.

[7] C. Gentry and A. Silverberg, "Hierarchical id based cryptography," in *Advances in Cryptology (ASIACRYPT 2002)*, vol. 2501 of LNCS, 2002, pp. 548–566.

[8] W. Tzeng, "A time-bound cryptographic key assignment scheme for access control in a hierarchy," *IEEE Trans. on Knowledge and Data Engineering*, vol. 14, no. 1, pp. 182–188, 2002.

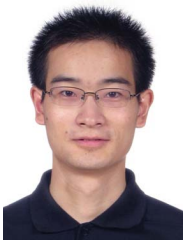
[9] D. Boneh and M. Hamburg, "Generalized identity based and broadcast encryption schemes," in *ASIACRYPT*, 2008, pp. 455–470.
[10] X. Liang, Z. Cao, H. Lin, and J. Shao, "Attribute based proxy re-encryption with delegating capabilities," in *ASIACCS*, 2009, pp. 276–286.
[11] B. W. D. Boneh, C. Gentry, "Collusion resistant broadcast encryption with short ciphertexts and private keys," in *Advances in Cryptology (CRYPTO'2005)*, vol. 3621 of LNCS, 2005, pp. 258–275.
[12] Y. Zhu, G. Ahn, H. Hu, and H. Wang, "Cryptographic role-based security mechanisms based on role-key hierarchy," in *ASIACCS*, D. Feng, D. A. Basin, and P. Liu, Eds. ACM, 2010, pp. 314–319.
[13] Y. Zhu, H. Hu, G. Ahn, H. Wang, and S. Wang, "Provably secure Role-Based encryption with revocation mechanism," *Journal of Computer Science and Technology*, vol. 26, no. 4, pp. 697–710, Jul. 2011. [Online]. Available: http://dx.doi.org/10.1007/s11390-011-1169-9
[14] L. Zhou, V. Varadharajan, and M. Hitchens, "Enforcing role-based access control for secure data storage in the cloud," *Comput. J.*, vol. 54, no. 10, pp. 1675–1687, 2011.
[15] E. Bertino, N. Shang, and S. Wagstaff, "An efficient time-bound hierarchical key management scheme for secure broadcasting," *IEEE Trans. on Dependable and Secure Computing*, vol. 5, no. 2, pp. 65–70, 2008.
[16] R. Sandhu, E. Coyne, H. Fenstein, and C. Youman, "Role-based access control models," *IEEE Computer*, vol. 29, no. 2, pp. 38–47, 1996.
[17] H. Chien, "Efficient time-bound hierarchical key assignment scheme," *IEEE Trans. on Knowledge and Data Engineering*, vol. 16, no. 10, pp. 1301–1304, 2004.
[18] X. Yi and Y. Ye, "Security of tzeng's time-bound key assignment scheme for access control in a hierarchy," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 4, pp. 1054–1055, 2003.
[19] X. Yi, "Security of chien's efficient time-bound hierarchical key assignment scheme," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 9, pp. 1298–1299, 2005.
[20] Y. Zhu, M. Yu, H. Hu, G. Ahn, and H. Zhao, "Efficient construction of provably secure steganography under ordinary covert channels," *SCIENCE CHINA Information Sciences*, vol. 55, no. 7, pp. 1639–1649, 2012.
[21] Y. Zhu, G. Ahn, H. Hu, S. Yau, H. An, and C. Hu, "Dynamic audit services for outsourced storages in clouds," *IEEE T. Services Computing*, vol. 6, no. 2, pp. 227–238, 2013.
[22] E. Goh, H. Shacham, N. Modadugu, and D. Boneh, "Sirius: Securing remote untrusted storage," in *Proceedings of the Internet Society (ISOC) Network and Distributed Systems Security (NDSS) Symposium*, 2003, pp. 131–145.
[23] R. S. Q. W. M. Kallahalla, E. Riedel and K. Fu, "Plutus: Scalable secure file sharing on untrusted storage," in *Proceedings of the 2nd USENIX Conference on File and Storage Technologies (FAST)*, 2003, pp. 29–42.
[24] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Advances in Cryptology (CRYPTO'01)*, vol. 2139 of LNCS, 2001, pp. 213–229.
[25] D. Boneh and H. Shacham, "Group signatures with verifier-local revocation," in *ACM Conference on Computer and Communications Security*, 2004, pp. 168–177.
[26] D. Boneh, X. Boyen, and E. Goh, "Hierarchical identity based encryption with constant size ciphertext," in *Advances in Cryptology (EUROCRYPT'05)*, vol. 3494 of LNCS, http://eprint.iacr.org/2005/015, 2005, pp. 440–456.
[27] D. Naor, M. Naor, and J. Lotspiech, "Revocation and tracing schemes for stateless receivers," in *Advances in Cryptology (Crypto'2001)*, vol. 2139 of LNCS, 2001, pp. 41–62.
[28] D. Boneh and B. Waters, "A fully collusion resistant broadcast, trace, and revoke system," in *ACM Conference on Computer and Communications Security*, 2006, pp. 211–220.
[29] Microsoft, "How encrypting file system works," Microsoft TechNet Report, 2009, http://technet.microsoft.com/en-us/library/cc781588(WS.10).aspx.
[30] SEC1, "Standards for efficient cryptograhy group: Elliptic curve cryptography," version 1.0, 2000, http://technet.microsoft.com/en-us/library/cc781588(WS.10).aspx.
[31] SEC2, "Standards for efficient cryptograhy group: Recommended elliptic curve domain parameters," version 1.0, 2000, http://technet.microsoft.com/en-us/library/cc781588(WS.10).aspx.

Yan Zhu received the PhD degree in computer science from Harbin Engineering University, China, in 2005. He was an associate professor of computer science at the Institute of Computer Science and Technology, Peking University in China from 2007 to 2012. He is currently a professor in the School of Computer and Communication Engineering at University of Science and Technology Beijing, China. He was a visiting associate professor in the Department of Computer Science and Engineering, Arizona State University From 2008 to 2009. He was a visiting research investigator of the Department of Computer and Information Science, University of Michigan-Dearborn in 2012. His research interests include cryptography, secure computation, and network security. He is a member of the IEEE.

Gail-Joon Ahn received the PhD degree in information technology from George Mason University, Fairfax, Virginia, in 2000. He is currently a full professor in the School of Computing, Informatics, and Decision Systems Engineering of Ira A. Fulton School of Engineering and the director of Security Engineering for Future Computing Laboratory (SEFCOM) at Arizona State University. 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 US National Science Foundation, the US National Security Agency, the US Department of Defense, the US Department of Energy, CISCO, Bank of America, Hewlett Packard, Microsoft, and the Robert Wood Johnson Foundation. He received the US Department of Energy CAREER Award and the Educator of the Year Award from the Federal Information Systems Security Educators Association. He is a senior member of the IEEE. 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.

Hongxin Hu received the PhD degree in computer science from Arizona State University, Tempe, AZ, in 2012. He is an assistant professor in the Department of Computer and Information Sciences at Delaware State University, where he leads the the Information Security and Privacy Lab (InSP). His current research interests include access control models and mechanisms, security and privacy in social networks, security in cloud and mobile computing, network and system security, and secure software engineering. He is a member of the IEEE.

Di Ma received the BEng degree from Xian Jiaotong University, China, the MEng degree from Nanyang Technological University, Singapore, and the PhD degree from the University of California, Irvine, in 2009. She is an assistant professor in the Computer and Information Science Department at the University of Michigan-Dearborn, where she leads the Security and Forensics Research Lab (SAFE). She was with IBM Almaden Research Center in 2008 and the Institute for Infocomm Research, Singapore in 2000-2005. She is a member of the IEEE.



Shangbiao Wang received the B.S. degree in Mathematics from the Peking University, Beijing, in 2007. Currently, he is a PhD student at Peking University. His research interests include cryptography and network security.