Cryptographic Role-based Security Mechanisms Based on Role-Key Hierarchy *

Yan Zhu[†‡] Gail-Joon Ahn[§] Hongxin Hu[§] Huaixi Wang[♭]

[†]Institute of Computer Science and Technology, Peking University, Beijing 100871, China
[‡]Key Laboratory of Network and Software Security Assurance (Peking University), Ministry of Education, China
[§]Laboratory of Security Engineering for Future Computing (SEFCOM),
Arizona State University, Tempe, AZ 85287, USA
[♭]School of Mathematical Sciences, Peking University, Beijing 100871, China
{yan.zhu,wanghuaixi}@pku.edu.cn; {gahn,hxhu}@asu.edu

ABSTRACT

Even though role-based access control (RBAC) can tremendously help us minimize the complexity in administering users, it is still needed 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 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. Also, a proof-of-concept prototype implementation and performance evaluation are discussed to demonstrate the feasibility and efficiency of our mechanisms.

Categories and Subject Descriptors

D.4.6 [**Operation Systems**]: Security and Protection—*Access controls,Cryptographic controls*

General Terms

Algorithm, Security, Theory

Keywords

Access Control, Role-based Cryptosystem, Role-Key Hierarchy, Pairing-based Cryptosystem

1. INTRODUCTION

Role-based access control (RBAC), as a proven alternative to traditional access control including discretionary access control (DAC) and mandatory access control (MAC), has been widely adopted for various information systems over the past few years [11]. Even though RBAC can tremendously help us minimize the complexity in administering users, it is still needed to realize the notion of roles at the resource level. In other words, RBAC systems need to control a user's access to resources as well as resource-level management based on roles. Consequently, in order to provide effective resource management, it is inevitable to adopt various cryptographic capabilities for managing resources in RBAC systems. However, the existing cryptographic schemes have several limitations to address above-mentioned features since those schemes cannot accommodate access control features of RBAC.

In distributed environments, we can leverage RBAC models to enforce fine-grained policies for sharing resources [7]. However, the current cryptosystems do not support such shared modes because the encryption/decryption keys cannot be recognized between RBAC systems. As a consequence, the resources should be re-encrypted when they are transferred into another domain. Obviously, it is necessary to design an efficient cryptographic mechanism compatible with corresponding access control systems.

Some new technologies, such as identity-based encryption (IBE) [3], attribute-based encryption (ABE) [8], and public-key broadcast encryption (PBE) [5] lay out a solid foundation for designing an efficient cryptosystem. Inspired by these techniques, in this paper, we propose a practical cryptographic RBAC model, called role-key hierarchy model, to support a variety of security features including signature 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 constructions can provide more efficient and flexible control than other hierarchical key assignments [1].

The rest of the paper is organized as follows. Section 2 overviews the role hierarchy in RBAC and Section 3 articulates our role-key hierarchy structure and RBC construction. In Section 4, we address our application schemes for the proposed role-key hierarchy and RBC. In Section 5, we briefly evaluate the performance of our schemes followed by the conclusion with our future work.

2. PRELIMINARIES

2.1 Partial Orders

*Supported by 863 Project of China (No.2006AA01Z434) and NSF of China (No. 10990011).

ASIACCS'10 April 13–16, 2010, Beijing, China.

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 $x \prec y$ 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 while $\downarrow x = \{y \in P | y \preceq x\}$ denotes the set of successors.

2.2 Role Hierarchy

In an information system, a hierarchy is used to denote the relationships and arrangements of the 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.

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. [10]:

DEFINITION 1. [Hierarchical RBAC model]*: The RBAC model has the following components:*

- *U, R, P, and S denote 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 $\preceq$;*
- *$user : S \to U$, a function mapping each session s_i to the single user $user(s_i)$; and*
- *$roles : S \to 2^R$, 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: $\bigcup_{r \in roles(s_i)} \{p \in P | \exists r'' \in R, r'' \preceq r : (p, r'') \in PA\}$.*

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 role hierarchy is shown in Figure 1, in which more powerful (senior) roles are shown toward the top of the diagram and less powerful (junior) roles toward the bottom.

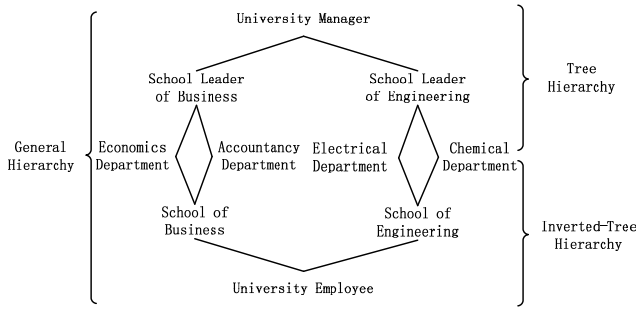

Figure 1: Example of role hierarchy with tree, inverted-tree, and general hierarchies.

Based on the specific features of resource management, we divide role hierarchy into three categories:

1. *Tree hierarchy*: It is useful to support the **sharing** of resources, in which resources make available to junior roles are also available to senior roles.
2. *Inverted-tree hierarchy*: It allows the **aggregation** of resources from more than one role, in which the senior can access resources in all subordinate roles.
3. *General hierarchy*: It can compose various different structures into a role hierarchy. Thus it facilitates both the **sharing and aggregation** of resources.

3. ROLE KEY HIERARCHY

3.1 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 is a cryptographic partial order relation for the sets of users, keys, and roles, denoted by $\mathcal{H} = \langle U, K, R, \preceq \rangle$, satisfying the following conditions:*

1. *$K = PK \cup SK$, the key set K includes the role-key set PK and the user-key set SK;*
2. *$UKA \subseteq U \times SK$, a one-to-many user to key assignment relation, i.e., each user $u_{i,j} \in U$ is assigned to an exclusive user-key $sk_{i,j} \in SK$;* [1]
3. *$RKA \subseteq R \times PK$, 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. *$KH \subseteq PK \times PK$, is a partial order on PK called the key hierarchy or key dominance relation, also written as $\preceq$; and*
5. *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 $(u_{i,j}, r_i) \in UA$.*

[1]In the notations $u_{i,j}$ and $sk_{i,j}$, i and j as subscripts represent the index variable of role and user, respectively.

where, $\langle K, \preceq \rangle$ is the smallest partially ordered set satisfying the above conditions. The user holds multiple user keys if he is a member of multiple roles in role hierarchy.

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.

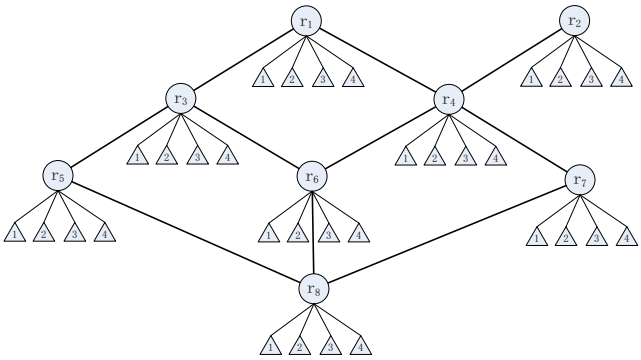

Figure 2: Example of role-key hierarchy

Our main objective is to map the role hierarchy in RBAC into a key management system. According to the condition 3 and 4, 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. Figure 2 shows an example of role-key hierarchy, in which the circle denotes the role key and the triangle denotes the user key, respectively. Note that this is a general hierarchy.

3.2 Role-based Cryptosystem

For ease of use, we expect that a 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.

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 $\mathcal{H} = \langle U, K, R, \preceq \rangle$ 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 secret key, 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: $params = params \bigcup \{lab_{i,j}\}$.

In public-key settings, a user does not hold any private information and 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.

3.3 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. The *derivation* function of a user's private key is forbidden even for the cases of partial order relations,

$$\Pr[Delegate(sk_{i,j}, r_l) = sk_{l,j'}] \leq \epsilon, \forall r_l \preceq r_i. \quad (1)$$

where, ϵ is small enough. Hence a user cannot use this capability to obtain new keys or identities.

In order to ensure system security, RKH also needs to satisfy 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 capability, there exists an efficient tracing algorithm to identify the corrupted users or gain the corresponding evidence.

The RKH is 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. This kind of attack involves the following cases:

- Collusion attack for framing users, in which the corrupted users in $\mathcal{R} = \{u_{i_k,j_k}\}_{k=1}^{t}$ wish to forge a new or unused key in $U \setminus \mathcal{R}$ (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 $R = \{u_{i_k,j_k}\}_{k=1}^{t}$ wish to forge a new or unused key in $R \backslash \{r_{i_1}, \cdots, r_{i_t}\}$. The aim of this attack is to change the privilege in partial order hierarchy.

It is a challenging task to avoid collusion attack since the traitors (corrupted users) have been granted users before they are detected. Traitor tracing is an efficient method to tackle this attack. However, we must ensure that the traitors cannot forge an 'unused' key to avoid tracing but leave some 'foregone' clue of evidence to discover them.

4. CRYPTOGRAPHIC SECURITY MECHANISMS BASED ON RKH

In this section, we present our role-based cryptosystem scheme with role-key hierarchy based on pairing-based cryptosystem. Also, applicable cryptographic mechanisms are proposed to demonstrate the feasibility and efficiency of our model and mechanisms.

4.1 Bilinear Pairings

We set up our systems using bilinear pairings proposed by Boneh and Franklin [2, 4]. Let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ be three cyclic groups of large prime order p. $\mathbb{G}_1$ and $\mathbb{G}_2$ are two additive groups and $\mathbb{G}_T$ is a multiplicative group using elliptic curve conventions. Let $\hat{e}$ be a computable bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$[2] 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.

4.2 Role-based Cryptosystem Scheme

Let $\mathcal{H} = \{U, K, R, \preceq\}$ is a role-key hierarchy with partial-order $\preceq$. Without loss of generality, we assume that the total number of roles is m in $\mathcal{H}$, i.e., $R = \{r_1, r_2, \cdots, r_m\}$. We construct our 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. We define

$$\begin{cases} D_i &= [\tau_i]G \in \mathbb{G}_1 \quad \forall r_i \in R, \\ V &= e(G, H) \qquad\quad \in \mathbb{G}_T. \end{cases} \tag{2}$$

  Each τ_i is called as the secret of a role and D_i is the identity of a role. Further, it defines $D_0 = [\tau_0]G$ by using a random $\tau_0 \in \mathbb{Z}_p^*$. Thus, public parameter is

$$params = \langle H, V, D_0, D_1, \cdots, D_c \rangle \tag{3}$$

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

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

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

$$\begin{cases} pk_i &= \langle H, V, W_i, \{D_k\}_{r_k \in \uparrow r_i} \rangle \\ W_i &= D_0 + \sum_{r_i \not\preceq r_k} D_k, \end{cases} \tag{4}$$

  where, $\{D_k\}_{r_k \in \uparrow r_i}$ 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 brevity, 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

$$\begin{cases} lab_{i,j} &= x_{i,j} &\in \mathbb{Z}_p^* \\ A_{i,j} &= \left[\frac{x_{i,j}}{\zeta_i + x_{i,j}}\right] G &\in \mathbb{G}_1, \\ B_{i,j} &= \left[\frac{1}{\zeta_i + x_{i,j}}\right] H &\in \mathbb{G}_2. \end{cases} \tag{5}$$

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 role keys are available in public.

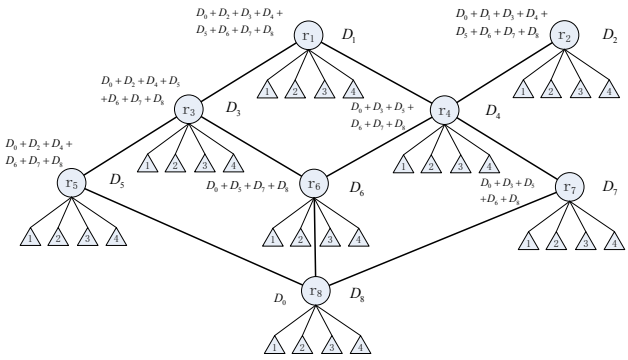

Figure 3: Example of role-key relationship on RBAC.

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 $r_i || r_j$, so that partial order relation in role keys can be defined as

$$\sum_{r_i \not\preceq r_k} D_k = \sum_{r_k \in Ind(r_i)} D_k + \sum_{r_k \in Succ(r_i)} D_k, \tag{6}$$

where, $Ind(r_i)$ and $Succ(r_i)$ denote the set of incomparable roles and successors for r_i, respectively. This is illustrated in Figure 3 (the top is senior-most roles and the bottom is junior-most roles), with the key representation of W_i on the left of the node and D_i on the right.

4.3 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:

DEFINITION 3 (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 $\mathcal{H} = \langle U, K, R, \preceq \rangle$ according to Setup and GenRKey algorithms in RBC model;*

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 purported 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, RL): $Verify(pk_i, \sigma, M) \rightarrow valid/invalid$;*

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:

$$Verify(pk_i, Sign(pk_i, sk_{i,j}, M), M) = valid. \quad (7)$$

- 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.
- Traceability: Given any valid signature, the manager or trusted third party should be able to trace who issued the signature by the user's secret key.
- 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.

4.4 Role-based Encryption

Encryption 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 [6] and Plutus [9]. 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 4 (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 $\mathcal{H} = \langle U, K, R, \preceq \rangle$ according to Setup and GenKey algorithms in RHC model;*

Encrypt: *Takes the encryption key pk_i and a plaintext M. It produces a ciphertext $\mathcal{C}$: $Encrypt(pk_i, M) \rightarrow \mathcal{C}_i$.*

Decrypt: *Takes the user key $sk_{i,j}$ and the ciphertext $\mathcal{C}$. 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:

$$Decrypt(sk_{i,j}, Encrypt(pk_l, M)) = M \quad (8)$$

where $r_l \preceq r_i$ and $(u_{i,j}, r_i) \in UA$.

Role-based encryption not only presents a solution for secure resource sharing across large-scale network, but also improves interoperation between different types of platforms. In addition, we believe it can also provide following security features:

- Protection against data leakage on the physical device, possibly caused by an untrusted administrator, a stolen laptop or a compromised server;
- Detection and prevention of unauthorized data modifications using a syncretic security mechanism based on policy-based access control and dynamic cryptographic technology;
- Changing users' access privileges by dynamically converging the information of users' decryption keys to generate one-time role-based encryption keys; and
- 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.

5. 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. As shown in Figure 4, 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

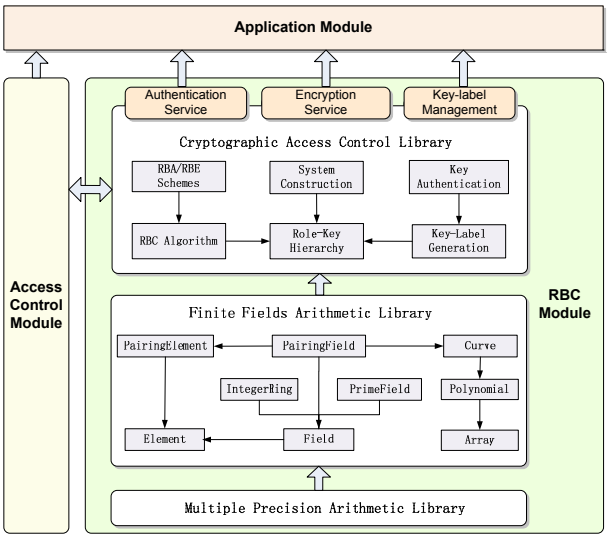

Figure 4: Cryptographic access control system based on role-key hierarchy.

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 [3] and key-label management services for the application module.

The experimental results show our constructions are able to provide better scalability, which is an important requirement for RBAC [10]. 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. Moreover, our constructions support a large-size of role hierarchy with arbitrary structures. Therefore, we believe our schemes can be applied to large-scale role-based cryptosystems, such as healthcare and financial systems.

Table 1: Parameters choosing under different scales.

Parameters	Small size	Medium size	Large size
number of roles	10's	100's	1000's
number of users	10-50	50-100	100-200
height of hierarchy	1-4	5-8	9-12
total number of users	100-1,000	1,000-10,000	10,000-100,000

We can consider several degrees to measure the scalability of our method as follows: 1) small scale (10's), 2) medium scale (100's), and 3) large scale (1000's). We estimate different parameters under different scales shown in Table 1, in which we assume that the size of relations is proportional to the size of roles.

6. 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 and encryption constructions on elliptic curve cryptosystem. 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.

[3]Note that the authentication service is beyond the scope of this paper.

7. REFERENCES

[1] 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*, 5(2):65–70, 2008.

[2] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In *Advances in Cryptology (CRYPTO'01)*, volume 2139 of LNCS, pages 213–229, 2001.

[3] D. Boneh and M. Hamburg. Generalized identity based and broadcast encryption schemes. In *ASIACRYPT*, pages 455–470, 2008.

[4] D. Boneh and H. Shacham. Group signatures with verifier-local revocation. In *ACM Conference on Computer and Communications Security*, pages 168–177, 2004.

[5] B. W. D. Boneh, C. Gentry. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *Advances in Cryptology (CRYPTO'2005)*, volume 3621 of LNCS, pages 258–275, 2005.

[6] 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*, pages 131–145, 2003.

[7] 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)*, pages 200–209, 2006.

[8] X. Liang, Z. Cao, H. Lin, and J. Shao. Attribute based proxy re-encryption with delegating capabilities. In *ASIACCS*, pages 276–286, 2009.

[9] R. S. Q. Mahesh Kallahalla, Erik 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)*, pages 29–42, 2003.

[10] R. Sandhu, E. Coyne, H. Fenstein, and C. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, 1996.

[11] 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)*, pages 47–63, 2000.