Role-Based Cryptography

P. Vijaya Sree¹, G. Praveen Babu²
¹Dept. of Computer Science & School of IT, Jawaharlal Nehru Technological University, India
²Dept. of Computer Science & School of IT, Jawaharlal Nehru Technological University, India
¹vijayasreepagolu@gmail.com

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.

Keywords: role-based access control, cryptography, encryption, role-key hierarchy, elliptic curve, role-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 [14]. 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. In distributed environments, we can leverage RBAC models to enforce fine-grained policies for sharing resources [10]. 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. In fact, 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.
2. Partial Orders

Let \( P \) 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 \( P \) are said to be comparable if \( x \preceq y \) or \( y \preceq x \). Otherwise, they are incomparable, denoted by \( x \not\preceq y \). 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 \( x \preceq d y \). In addition, if \( x \prec y \) and \( x \preceq z \prec y \), 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 \( P \) as follows: For an element \( x \) in \( P \), \( \uparrow x = \{ y \in P \mid y \preceq x \} \) denotes the set of predecessors of \( x \), and \( \downarrow x = \{ y \in P \mid y \preceq x \} \) denotes the set of successors.

3. 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 \( hR, i \) in RBAC, role-key hierarchy is a cryptographic partial order relation for the sets of users, keys, and roles, denoted by \( H = hU, K, R, i \), 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-one user to key assignment relation, i.e., each user \( u, i \in U \) is assigned to an exclusive user-key \( sk_{i,j} \in SK \);
3. \( RK A \subseteq R \times PK \), 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 \( _i \); and
5. Each user \( u, i \) can access the resources associated with \( rl \) if and only if \( rl \preceq r_i \in RH \) and \( (u, i, r_i) \in UA \), where \( hK, i \) is the smallest partially ordered set satisfying the above conditions. The user holds multiple user keys if he is 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 cryptographic algorithms, such as Encrypt and Decrypt, which can realize various access control functions by using role keys and user keys in role-key hierarchy.

4. Role-based Cryptosystem

We expect that a system manager assigns the user key \( sk_{i,j} = (lab_{i,j}, dki_{i,j}) \) to a user, where \( lab_{i,j} \) is a public label and \( dki_{i,j} \) 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 \( hR, i = hR, i \) and a security parameter \( s \), Role-based Cryptosystem (RBC) is a key
management system that can construct a role-key hierarchy $H = h(U.R, i \rightarrow)$ and generate all keys on $H$, which is specified by three randomized algorithms, Setup, KeyRGen, and AddUser, described as follows:

- Setup($s,n,m$): Takes a security parameter $s$ and a role hierarchy $H$ as input. It produces a manager key $mk$ and an initial parameter params, that is, $\text{Setup}(s) \rightarrow \{H, mk, \text{params}\}$. 
- GenRKey(params, $ri$): Takes the parameter params and a role index $ri$. It generates a role key $sk_i$ in $ri$, that is, $\text{KeyRGen}(\text{params}, ri) \rightarrow sk_i$
- AddUser($mk$, ID, $ui,j$): Takes a user identity ID, a user index $ui,j$, 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 $ui,j$, that is, $\text{AddUser}(mk, ID, ui,j) \rightarrow sk_{i,j} = (lab_{i,j}, dk_{i,j})$. The user label $lab_{i,j}$ is added to the public encryption key.

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.

### 5. 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 $ui,j$ 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, $\Pr[\text{Delegate}(sk_{i,j}, cl) = sk_{i,j}’] \leq \phi, \forall cl \subseteq cl$. (1) where, $\phi$ 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 another’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;
- 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 $R = \{ui,k,|k\}$ $k=1$ wish to forge a new or unused key in $U \setminus R$ (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 $R = \{ui,k,|k\}$ $k=1$ wish to forge a new or unused key in $R(\rightarrow r_1, \cdots, \rightarrow r_t)$. 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 A. 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. The number of colluders $|R| = t$ is an important parameter. A RBC scheme is to be $(t,n,m)$-collusion secure if for any subset of $t$ in $R$ with $|U| = n$ and $|R| = m$, the adversary can gain the advantage from $R$ to break this scheme. It is said to be fully collusion secure when it is $(n,n,m)$-collusion secure.

### 6. 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.
7. 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 [9] and Plutus [12]. However, these systems implement some trivial schemes where the number of cipher text 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.

8. 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.

9. Acknowledgement

This work is supported and made under the esteemed guidance of G. Praveen Babu garu, who is a professor in software engineering department at School of IT, Jawaharlal Nehru Technological University, Hyderabad.

REFERENCES


