CRYPTOGRAPHIC KEY MANAGEMENT FOR ROLE-BASED ACCESS CONTROL MODEL IN POWER SYSTEM COMPUTER NETWORKS

by

CELIA J LI

Bachelor of Science, Tianjin, P.R. China, 1998

A thesis
presented to Ryerson University
in partial fulfillment of the
requirement for the degree of
Master of Applied Science
in the Program of
Electrical and Computer Engineering

Toronto, Ontario, Canada, 2005

Author's Declaration

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Signature: ______________________________

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Signature: ______________________________

Instructions on Borrowers

Ryerson University requires the signatures of all persons using or photocopying this thesis.

Please sign below, and give address and date.

Name	Signature	Address	Date

CRYPTOGRAPHIC KEY MANAGEMENT FOR ROLE-BASED ACCESS CONTROL MODEL IN POWER SYSTEM COMPUTER NETWORKS

CELIA J LI

Master of Applied Science
Electrical and Computer Engineering
Ryerson University
Toronto, Ontario, Canada, 2005

Abstract

This thesis research has successfully completed two developments: an efficient Power-system Role-based Access Control (PRAC) and a secure Power-system Role-based kEy Management (PREM). The PRAC significantly increases the security of computer networks for power systems, and surmounts the challenges caused by typical security and reliability concerns due to current technological and political changes faced in the electricity power industry. The PREM is designed to support the efficient operation of the PRAC using one-way hash functions and utilizing their advantages of computationally efficient and irreversibility security. PRAC and PREM are not only developed for handling single local computer network domain, but also extended for supporting multiple computer network domains. A platform for the comprehensive assessment of PREM is established for the fast and economical assessment of the key management developed in this thesis research.

Acknowledgments

I would like to express my deepest gratitude to my supervisor Professor Richard Cheung for his expert supervision. I highly appreciate the valuable advices, professional guidance, and consistent encouragement from Professor Cheung through all stages of my study and research.

I am grateful to Dr. David Xu and Dr. Lian Zhao for reviewing my thesis and providing advices.

I would like to dedicate this thesis to my family for their deepest love, remarkable encouragement and unconditional support.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

In the recent few years, North American electricity utilities have been undergoing substantially changes due to technological and political reasons. In the technological aspects, there has been rapidly increasing use of intelligent electronic devices (IEDs) with communication capabilities in the electricity power systems. These IEDs can communicate with the computer network directly or through a PC. In the political aspects, many electricity utilities have been under tremendous pressure from their government to provide their customers or third parties with open access to their transmission systems. Some large monopoly-type utilities have been forced to break up into smaller companies, typically one company to be responsible for generation, one for transmission, and one for distribution. These companies, however, must operate very closely through computer communications, since they share one physical electricity circuit network. Regardless technological or political reasons, the needs for computer network communications in electricity power system have been increasing dramatically [1].

The security and reliability concerns of the computer network in the electricity power system urgently need to be addressed [2-4], as the demands for access to the network have been rising rapidly and tremendously due to several key reasons. First, under the new government open access policy imposed on the utilities, the power system computer networks, originally isolated and used by qualified staff, have to become more accessible to a wider range and number of users that increases the dangers of the network to be attacked. Second, with the increasing use of IEDs of independent communications to the computer networks, the power system can be operated, on one hand, very efficiently if all network communications are absolutely correct, but the power system operation can be endangered, on the other hand, if there is any network error. Third, for companies broken up from a large utility under government open access policy, any mismatched command requests from these companies for operating the same physical electricity circuit will jeopardize the reliability of the power systems.

The prime objective of this thesis research is to develop a new network access control to significantly increase the security of computer networks for power systems. This network access control can surmount the challenges caused by the above-mentioned security and reliability concerns due to current technological and political changes faced in the electricity power industry.

This thesis has completed an exhaustive investigation on the currently available techniques, models, and managements for computer network access control. The existing technologies are logical and useful for many applications such as for banking systems, for insurance systems, for medical systems, etc. However, these techniques, models, and managements do not efficiently satisfy the requirements for reliable operations in the electricity power systems. This initiates the research in this thesis.

Initiated from and built on currently available network access and security technologies, this thesis research extends significantly the capacity of the conventional role-based access control for applications in electricity power systems. A new network access architecture is developed based on the extended role-based access control model. Furthermore, built on the existing key management for network access control, this thesis research designs an efficient key management using a simple one-way hash function to control the access to the power system computer network and the execution of the power system network controlling commands.

1.2 Introduction of Power System Computer Network Access Control

The network access control architecture, designed in this thesis research, is comprised of a host domain for the utility being studied and multiple foreign domains for neighboring utilities. The host domain contains a central domain for the power system control center network and several local domains for computer networks of generation systems, transmission systems, distribution systems, and customer loads.

The control center network is designed in this thesis to operate as an administrator for all local domains in the host utility, and also communicates with users from neighboring utilities using the direct communication link or through the Internet. The control center defines the security policies for all local domains in the host domain as well as for connections with the foreign domains, and authorizes privileges of access to foreign domain users. Each local domain (either for generation systems, transmission systems, distributed systems, or customer loads) implements its security policy instrumented by the control center. The local domain security officer authorizes privileges of access to its own local domain users and the users from other interconnected local domains.

A new role-based access control model, developed in this thesis research, is different from the conventional models. The new model defines the relationships between network roles in a local domain and those of interconnected local domains. Also the new model extends the access control from one power enterprise domain to foreign enterprise domains.

In the new model developed in this thesis research, a role is defined as a collection of privileges of network access that can be executed by the authorized users. A role can take on a number of privileges, a user can be assigned with a number of roles, and a role can be assigned to multiple users. In general, a set of roles can be assigned to a particular user according to the user's responsibility and authority in the power enterprise. A privilege in the new model is an access that can be exercised on objects, such as monitoring power system performance, trading electricity, operating substation equipments, etc.

With the architecture created in this thesis research, the network control center is designed to have the authority and facilities to structure the enterprise domain role hierarchy. This design allows the control center computer network to have full capability to structure the network access according to the real power system environment as well as capable of adapting to the changes in the environment. The enterprise hierarchy can be comprised of multiple local-domain role hierarchies. A role hierarchy is established to represent inheritance of authority, responsibility, and privilege among the roles.

The network controller in the new model, developed in this thesis research, is responsible to evaluate constraints to the role operations. The new model is designed to efficiently handle different types of role constraints. The role constraints in general can be grouped into three types: cardinality constraint, separation-of-duty constraint and prerequisite constraint.

The cardinality constraints include that a role may be allowed to have a limited number of users; a user may be allowed to execute a limited number of roles, etc. The separation-of-duty constraints enforce conflict-of-interest prevention policies established by the control center for the access control. Conflict of interest arises as a result of the simultaneous assignment of two mutual exclusive roles to the same user. For example, a role of network monitor and a role of network operator cannot be assigned to the same user in order to avoid the conflict of interest in power enterprise systems.

The prerequisite constraints for specific network accesses are checked by the network controller. The controller will ensure that a user can perform a prerequisite operation if and only if the user is already a member of prerequisite role. For example, a role with a privilege to initiate a tripping command to the substation bus-tie breaker is a role that has a prerequisite constraint. The constraint is determined by the control center such as the role must be a substation operator with a specific training and experience. A user can execute this role if and only if the user already has a role of substation operators. Moreover, there exist many other role constraints, for instance time constraint that defines how long the role can be activated.

1.3 Introduction of Contents in this Thesis

The following provides an introduction of the chapters in this thesis.

Chapter 2 presents the cryptographic support for the computer network security, two typical types of existing encryption and decryption techniques, the features of hash functions for the network security, the conventional computer network access control, and the key management for the network security. This chapter outlines the research initiations. One initiation was built on currently available network access and security technologies to extend the capacity of the conventional role-based access control for applications in electricity power systems. Another initiation was built on the existing key management for network access control to design an efficient key management.

Chapter 3 presents a security management architecture and a new role-based access control model specially designed for power system computer networks. This model extends the capability of the conventional models for handling multiple computer network domains. This chapter develops a security policy system and illustrates the implementation of this system with examples. Finally this chapter presents a procedure for assigning the privileges to the roles in the new model for the power system applications, and illustrates the procedure with three cases.

Chapter 4 presents a new key management for electricity power system computer networks. This key management uses one-way hash functions and utilizes their advantages of computationally efficient and irreversibility security. This chapter presents four core components in the new key management that include rules for key generation, key management for dynamic hierarchy, algorithms for key generation and key modification, and procedure of object access using keys. This chapter has shown that the new key management is extended from one power system computer network local domain to the multiple local domains. The extended PREM decentralizes the key management for each local domain that is independently managed by its own local domain administrator. Any change of the role hierarchy structure in one local domain does not affect the keys of other local domains.

Chapter 5 presents a platform for the comprehensive assessment of PREM for the role-based access control of power system computer networks. Since in practice it is very difficult particularly within the university environment, to thoroughly assess the network access control, this chapter proposes a simple alternative that consists of two stages. The first stage for functionality assessment of PREM uses two typical cryptographic algorithms: one for one-way hash function, one for symmetric key encryption. The second stage for benchmark assessment of PREM is via the comparison using typical existing key management method. Three study cases are given to illustrate the assessment.

Chapter 6 presents the conclusion of this thesis research and the future work in power system computer network access control.

Chapter 2

Network Security and Access Control

This chapter first investigates the cryptographic support for the computer network security. The network security is one of the most important components for the reliable operations of the entire electricity power system. Second, this chapter discusses two typical types of existing encryption and decryption techniques. Third, this chapter assesses the features of hash functions for the network security. Fourth, this chapter investigates the conventional computer network access control. Finally, this chapter introduces the key management for the network security.

The currently available techniques, models, security managements are useful for many applications such as for banking systems, for insurance systems, for medical systems, etc. However, they do not efficiently satisfy the requirements for reliable operations in the electricity power systems. This initiates the research in this thesis.

This chapter outlines the research initiations. One initiation was built on currently available network access and security technologies to extend the capacity of the conventional role-based access control for applications in electricity power systems and to develop a new network access architecture based on the extended role-based access control model. Another initiation was built on the existing key management for network access control to design an efficient key management using a simple one-way hash function to control the access to the power system computer network and the execution of the power system network controlling commands.

The following lists the sections in this chapter:

Section 2.1 presents the basic concepts in cryptography.

Section 2.2 discusses typical techniques of encryption and decryption for network security.

Section 2.3 explores features and best uses of hash functions for network security.

Section 2.4 reviews conventional network access controls.

Section 2.5 assesses key managements for network security.

Section 2.6 summarizes research review findings and thesis research initiations.

2.1 Cryptographic Support for Computer Network Security

This section discusses the cryptographic support for computer network security. Cryptography is defined as the study of mathematical techniques related to aspects of information security such as confidentiality, data integrity, entity authentication, and data origin authentication [5]. The use of cryptographic techniques not only provides information security but also offers: c*onfidentiality* ensuring that no one can read the message except the intended receiver, *data integrity* assuring the receiver that the received message has not been altered, *authentication*, proving the identity of the user, and n*on-repudiation* proving that the sender accurately has sent the message.

A goal of cryptography is to adequately address the above four areas in both theory and practice. Cryptography is about the prevention and detection of cheating and other malicious activities.

In data transmissions, the use of cryptography is necessary when communicating over any untrusted medium that includes any network particularly the Internet. Traditionally, two important mechanisms of cryptography such as encryption and decryption are widely used. A message in its original form is known as plaintext or cleartext. The mangled information is known as ciphertext. The process for producing a ciphertext from a plaintext is known as encryption. The reverse of encryption is called decryption that converts a ciphertext back into its original cleartext.

Cryptographic process uses an encryption/decryption algorithm and a secret value known as the key to increase the data security. With a good cryptographic scheme, it is fine to have everyone know the algorithm used in the encryption because the knowledge of the algorithm without the key virtually could not decrypt the information.

2.2 Encryption-Decryption Techniques for Network Security

This section reviews the encryption and decryption techniques for enhancing the security of the computer networks. Basically the techniques for the computer network encryption can be grouped into two types: symmetric encryption and asymmetric encryption. The symmetric encryption involves the use of a single key. Given a message (called the plaintext) and the key, the encryption produces a ciphertext, which is about the same length as the plaintext was. The decryption is the reverse of encryption and uses the same key.

The symmetric encryption process can best be characterized as follows: (1) Encryption turns a clear text into a ciphertext, (2) Decryption restores the clear text from the ciphertext, and (3) The same key is used in both encryption and decryption.

A) Symmetric Encryption

The three common symmetric encryption algorithms available in commercial devices are Data Encryption Standard (DES), Triple Data Encryption Standard (3DES), and Advanced Encryption Standard (AES) [6].

DES is one of the most widely used standards. The DES turns a cleartext into a ciphertext via an encryption algorithm. The decryption algorithm on the receiving end restores the cleartext from the ciphertext. Keys are used in the process of encryption and decryption. The most widely used symmetric DES scheme operates on 64-bit message blocks and uses 64-bit keys.

3DES is an enhancement to DES that preserves the existing investment in software but makes a brute-force attack more difficult. The 3DES performs the operations of encryption and decryption using one, two, or three different keys.

AES is the newest encryption algorithm. It currently specifies keys with a length of 128, 192, or 256 bits to encrypt blocks with a length of 128, 192, or 256 bits.

It is important that if a decision needs to be made in the cryptographic algorithm development, then the decision should avoid the security, resulted from the use of the algorithm in data transmission, being compromised. In most cryptographic protocols, the cornerstone to security lies in the secrecy of the key used for the data encryption.

Symmetric encryption algorithms are designed so that it is extremely difficult for anyone to obtain the cleartext without the key.

B) Asymmetric Encryption

Asymmetric encryption is often referred to as public key encryption. This encryption includes three commonly-used algorithms: the Rivest-Shamir-Adelman (RSA) algorithm, the Diffie-Hellman algorithm, and the El Gamal algorithm.

Unlike the symmetric encryption algorithms, different keys are used in the asymmetric encryption and decryption processes. The sender and the receiver, each has one pair of keys: one private key that is not exposed to anyone, and one public key that is preferably to be known in public. The public key and the private key are different, but they are related as a pair such that the private key can decrypt any message encrypted by the public key. If anyone wants to communicate using the asymmetric encryption, each individually needs a specific pair of public key and private key made according to a commonly agreed key generation algorithm. For example, if Alice wants to send a message to Bob, she will encrypt the message using Bob's public key, and Bob can decrypt the message using his private key.

During the asymmetric encryption process, the private key is known only to the intended receiver, the public key is known to the public, and the public key distribution is not a secret operation.

Rivest-Shamir-Adelman (RSA) Cryptosystem is a public key cryptographic system that requires the use of a public key and a private key [6]. The following steps illustrate the key generation algorithm for RSA: First choose two large prime numbers of approximately the same size, namely p and q. Second compute the product of these two primes, n = p q. Third compute the value of function φ (n) = (p-1) (q-1). Fourth choose an integer e between 1 and φ(n) such that gcd (e, φ (n)) = 1. Finally compute d whereby $d = e^{-1}$ mod (φ (n)).

The public key is (n, e) whereas the private key is (n, d). Presently, 512-bit prime numbers are used, resulting in a 1024-bit value of n. Cracking such RSA systems require the ability to factor 1024-bit integers, which at the present time is almost impossible. However, RSA requires a lot of processing time that may decrease throughput performance.

The RSA encryption proceeds by transforming M, the digital form of the plaintext (the original message) into C, the ciphertext (the encrypted message) using the following function.

$$C = M^{e} \bmod n \qquad (2.1)$$

An inverse function is used to decrypt M.

$$M = C^{d} \bmod n \qquad (2.2)$$

2.3 Hash Function for Network Security

Data integrity is very important for data transporting over the public Internet, where the data could be intercepted and modified. Many methods are available to ensure the data integrity against a message being modified. A simple method is to send the message with an attached hash value that can be used at the receiving end to identify whether the message has been altered.

A hash value is used to verify the contents of a transmission that are the same at both ends of the transmission, similar to a checksum. A hash value is calculated using a hash function that takes any size input, such as a packet, and returns a fixed-size string. The hash value verifies the integrity of the original message. If the transmitted hash value matches the received hash value, the message has not been tampered with. However, if the two hash values do not match, the message has been altered.

A hash function H is a transformation that takes a variable-size input "m" and returns a fixed-size string, which is called the hash value "h", that is h = H (m). Examples of well-known hash functions are Message Digest (such as MD2, MD5) and Secure Hash Algorithm (SHA) [7]. Hash functions with the feature of variable-size input to fix-size output have a variety of general uses for data integrity protection. However when employed in cryptography, the hash functions with additional features are chosen.

The basic requirements for a cryptographic hash function H(x) are:

- The input x can be of any length,
- The output h = H(x) has a fixed length,
- H (x) is relatively easy to compute for any given x ,
- H (x) is one-way,
- H (x) is collision-free.

A hash function H is said to be one-way if given a hash value h, it is computationally infeasible to find the input x. For any two different messages x and y, if H (x) ≠ H (y), then H is said to be a collision-free hash function.

2.4 Conventional Computer Network Access Control

Access control to the computer network is comprised of three categories: Mandatory Access Control (MAC), Discretionary Access Control (DAC), and Role-based Access Control (RBAC). The MAC enforces the access control according to the information security labels attached to users and objects. The MAC can determine the kind of consistent access between a user and its objects. Security labels have to be granted to all users and objects by the network system officer, and the label can be changed only in accordance with the content of the object. The MAC policy is not flexible, so that it is not suitable to be used in commercial areas. In DAC, each object has an access control list, indicating that all the accesses to the users are authorized on that object. However, in a large distributed system there are thousands of objects, and each of which could be assigned to hundreds of users, so that the access control list will be enormous in size and its maintenance will be difficult and costly.

A) Fundamental Role-Based Access Control

To provide an improved access control to the computer network, the fundamental Role-Based Access Control (RBAC) was proposed [8-12]. The RBAC determines the access privileges that roles can perform, and assigns roles to users. Users can access objects according to the roles assigned to them. The central notion of the RBAC is that users do not directly access to enterprise objects. Instead, the access privileges are associated with roles, and each user is assigned to one or multiple members of appropriate roles. The fundamental RBAC model consists of three basic components: users, roles, and privileges, whereby a user is a human being or an autonomous agent, a role is a collection of privileges needed to perform a certain job function within an organization, and a privilege is an access mode that can be exercised on objects in the system. A user can be a member of many roles as determined by his/her responsibilities and qualifications, and a role can have multiple members. One role may have many privileges, and the same privilege can be associated with many roles. The roles can be easily reassigned to the users without modifying the underlying access structure.

B) Extended Role-Based Access Control

In the recent few years, several role-based access control models have been proposed to meet the security requirements for different applications. In addition to the fundamental role based models, a role-based access control administration model was proposed [13] to extend the basic one by adding the administrative roles and administrative privileges, which are dedicated to the management of the roles in the network. The model represents and formalizes complex management and delegation rules that an organization may have in place. However, the requirements that have driven the design of the access control system do not justify adding this degree of complexity solely to the management of granting and revoking roles. Therefore, the model is not suitable for electricity power system applications.

Role Graph Model (RGM) was proposed [14, 15] based on the notion of users, privileges and roles. In RGM, a privilege is viewed as the combination of an object and a set of operations on that object. More precisely, a privilege is a pair (x; m) defined in RGM, where x refers to an object and m is a non-empty set of access modes for the object x. A role is defined as a named set of privileges and is represented by a pair (rname;

rpset), where rname is the name of the role, and rpset represents the set of privileges of that role. Given a role R, it uses R:rname and R:rpset to refer to the role's name and privilege set respectively. The roles form the nodes of the role graph. An edge of R1 → R2 in the role graph defined in RGM represents that R1 is junior to R2 if and only if R1:rpset is a subset of R2 :rpset.

There are three factors that determine the network access control: (1) the assignment of privileges to roles, (2) the assignment of roles to users, and (3) the orientation of the edges in the role graph. The RGM model includes a MaxRole and a MinRole. The MinRole represents the minimum set of privileges available to all roles. The MaxRole represents the combination of all the privileges of the roles in the role graph. It does not need to have any users authorized to it. It is in the role graph in order to have a place to summarize all of the privileges in the network system.

The models presented above are all logical and useful for applications such as for banking systems. These models, however, do not efficiently satisfy the requirements of the electricity power systems. This initiates the research in this thesis.

2.5 Key Management for Network Security

This section reviews the previous work in the key management, which is important in information security that ensures that any access to an object is authorized.

A case was considered in [16, 17], where the users were divided into a number of disjoined sets, $S=\{S_1, S_2, \ldots.., S_n\}$. Each set S_i was designated as a node U_i in the graph, and each user was assigned to a security class called the security clearance. The nodes were partially ordered by the relationship of $\leq$. For example, $U_i \leq U_j$ meant that users in S_i

at node U_i had a security clearance lower than those in S_j at node U_j. In other words, users in S_j with higher security clearance can have access to information destined to users in S_i, but the opposite access was not allowed.

Inheritance relationships among the nodes in a hierarchy were defined. The relationship of $U_i \leq U_j$ was that U_i was defined as a direct child node of U_j, and U_j was defined as a direct parent node of U_i. The inheritance relationship was defined to be transitive, that is, if node U_i was the child node of U_j and U_j was the child node of $U_{k,}$, then U_i was called as an indirect child node of U_k, and U_k was called as an indirect parent node of $U_{i.}$ An example of a hierarchy was shown in Figure 2.6 where node A was a direct parent node of B and C, node B was a direct parent node of D, E, and F. Therefore, node A was an indirect parent node of D, E, F, G and H.

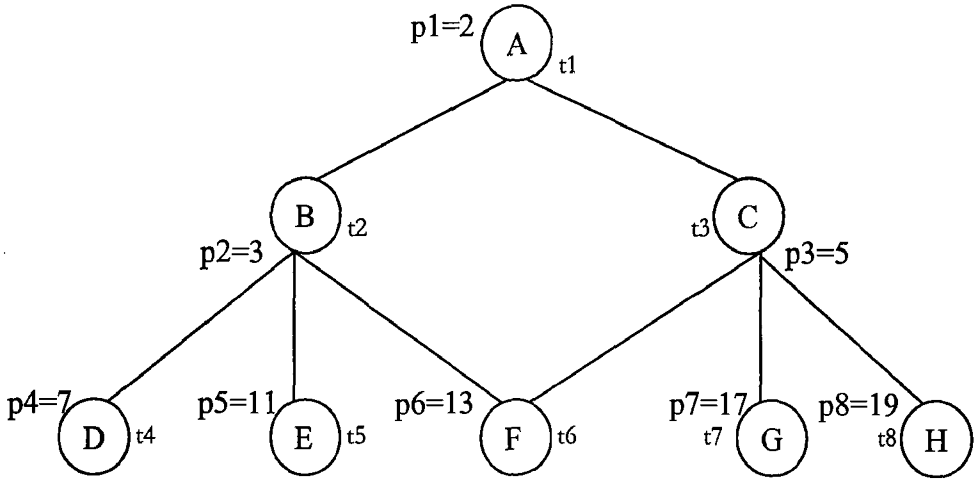


Figure 2.1: Inheritance Relationships Among Nodes in a Hierarchy

The central authority (CA) is responsible for the generation and distribution of keys to all nodes. Any node must have a correct key for accessing an object in another node, and any node can derive the keys of its direct or indirect child nodes. For example, if node X wants to access an object of node Y, then node X must be the direct or indirect parent of node Y, and node X must derive the key of node Y using the same scheme as the original key for node Y generated by CA. The following shows a scheme proposed on how to generate or derive the keys [16, 17].

A public integer or parameter t_i was assigned to each node U_i with the property represented by t_i/t_j , if and only if $U_i \leq U_j$. The CA chose a random key K_0 and a secret pair of prime number p and q. The product M=p*q was made public, then K_i=$K_0{}^{ti}$ mod M was distributed to the users in U_i. If $U_i \leq U_j$, then t_i / t_j was an integer and U_j can compute K_i by the following hash function.

$$K_i = K_0{}^{ti} \bmod M = (K_0{}^{tj})^{ti/tj} \bmod M = K_j{}^{ti/tj} \bmod M \qquad (2.3)$$

Notice from the hash function (2.3) that the key K_i can be computed from K_j. However, K_j cannot be computed from K_i . This computation was considered infeasible.

To accomplish the key generation scheme, each U_i is assigned with a small prime number p_n by the CA, and t_i can be computed by the following equation:

$$t_i = \prod_{U_n \sim\leq U_i} p_n \qquad (2.4)$$

where $U_n \sim\leq U_i$ means that U_n was not below U_i in the hierarchy. In other words, the computations of t_i indicated all the nodes which were not below U_i .

A simple example of the hierarchy is shown in Figure 2.6. To make the computation simple, the CA assigned the prime numbers for the p_n's. For example if the CA started with the smallest prime number, then node A was assigned with p1 = 2, node B with p2 = 3, node C with p3 =5, etc. According to (2.4), the public parameter t_i assigned to each node was the product of the prime numbers associated with nodes that were not below it in the hierarchy. Since node A does not have any nodes that were above it, its public parameter t_1 is assigned to be 1. The parameter values of all other nodes can be computed based on (2.4) as follows:

node A: $t_1 = 1$

node B: $t_2 = p1*p3*p7*p8 = 2*5*17*19=3230$

node C: $t_3 = p1*p2*p4*p5 =2*3*7*11 =462$

node D: $t_4 = p1*p2*p3*p5*p6*p7*p8 = 2*3*5*11*13*17*19=1385670$

etc.

The advantage of this scheme is that the key generation algorithm is quite simple. However, it has the following drawbacks. First, the key generation uses one hash function that only deals with one hierarchy structure. Second, prime numbers assigned to the nodes must be chosen carefully. Otherwise there exists the possibility of some users collaborating to compute a key to which they are not entitled. Finally, if a node is added to or deleted from the system, then all the keys need to be re-generated.

In an attempt to improve the key management, a method named "Optimal key assignment algorithm" was proposed [18]. In this method, the size of the integer assigned to each node was proportional to the number of the nodes in the hierarchy. This method would be difficult to implement when the number of nodes becomes very large. Nowadays, there are other solutions for the key management on multilevel security [19, 20], such as hierarchical key management scheme for XML data [21, 22] and for secure group communications [23-25]. These schemes have different drawbacks, such as some cannot efficiently provide the dynamic access control or some require huge storage for public parameters.

2.6 Research Findings and Initiations

This chapter has reviewed the available cryptographic support for the security of computer networks, and has discussed two typical types of existing encryption and decryption techniques. This chapter has assessed the features of hash functions for the network security, and has investigated the conventional computer network access control. Finally, this chapter has introduced the key management for the network security.

The currently available techniques, models, security managements discussed in this chapter are all logical and useful for many applications such as for banking systems, for insurance systems, for medical systems, etc. However, these techniques, models, and managements do not efficiently satisfy the requirements for reliable operations in the electricity power systems. This initiates the research in this thesis.

Initiated from and built on currently available network access and security technologies, this thesis research extends significantly the capacity of the conventional role-based access control for applications in electricity power systems. A new network access architecture is developed based on the extended role-based access control model, of which the details are given in Chapter 3.

Furthermore, built on the existing key management for network access control, this thesis research designs an efficient key management using a simple one-way hash function to control the access to the power system computer network and the execution of the power system network controlling commands. The details of the key management design are given in Chapter 4 and the design validation is given in Chapter 5.

Chapter 3

Power-system Role-based Access Control (PRAC)

This chapter details a new access control designed in this thesis research for electricity power system computer networks. This new control extends significantly the currently available role-based access controls to have advanced capacities for providing security controls of accesses to the power system computer networks. This control is named as Power-system Role-based Access Control (PRAC) in this thesis.

First this chapter presents a security management architecture specially designed for power system computer networks. This architecture is comprised of an electricity enterprise host domain and multiple foreign electricity enterprise domains that include computer network domains of electricity companies from other provinces and neighboring countries. The host domain contains a central domain for the power system control center computer network and several local domains for computer networks of generation systems, transmission systems, distribution systems, and customer loads. The main functions of the network domains in this specially designed architecture are discussed in this chapter.

Second this chapter presents the PRAC model that is a specific role-based access control model developed for the power system computer networks. This model differs from the conventional ones by extending the access control from one domain to multiple domains.

Third this chapter establishes an XML-based PRAC security policy system for the electricity power system enterprise domain and for connections with the foreign computer network domains. Examples to illustrate this XML-based policy system are given.

Finally this chapter presents a procedure for assigning the privileges to the roles in the PRAC model for the power system applications. Three study cases are given to illustrate the procedure.

The following lists the sections in this chapter.

Section 3.1 presents a security management architecture specially designed for the power system computer networks.

Section 3.2 presents the PRAC model, a specific role-based access control model developed for the power system computer networks.

Section 3.3 presents an XML-based PRAC security policy system and the administration of the policy.

Section 3.4 presents a privilege assignment method for PRAC model.

Section 3.5 provides the concluding remarks of this chapter.

3.1 Security Management Architecture

The security management architecture designed in this thesis research for electricity power system computer networks is comprised of an enterprise domain and multiple foreign enterprise domains that include computer network domains of electricity enterprises from other provinces or countries. The enterprise domain contains a central domain for the power system control center computer network and several local domains for computer networks of generation plants, transmission systems, distribution systems, and customer loads. Figure 3.1 shows the network security management architecture designed in this thesis research. In this architecture, the electricity enterprise network domain contains one central domain for the power system control center and four local domains for generation systems, transmission systems, distribution systems, and customer loads.

The control center computer network domain operates as an administrator of all local domains in the electricity enterprise, and also communicates with users of other electricity enterprises using the direct communication link or through the Internet.

The main functions of the network domains in the security management architecture designed in this thesis are as follows:

- The control-center network manager defines and maintains the security policies for all local domains in the enterprise computer network domain. Also, the control-center network manager defines and maintains the security policies for connections with the foreign domains and authorizes privileges of access to foreign domain users according to the defined foreign domain security policies.
- Each local domain (for generation systems, transmission systems, distributed systems, or customer loads) implements its security policy instrumented by the control center network manager. The local domain security officer authorizes privileges of access to its own local domain users and the users from other interconnected local domains.

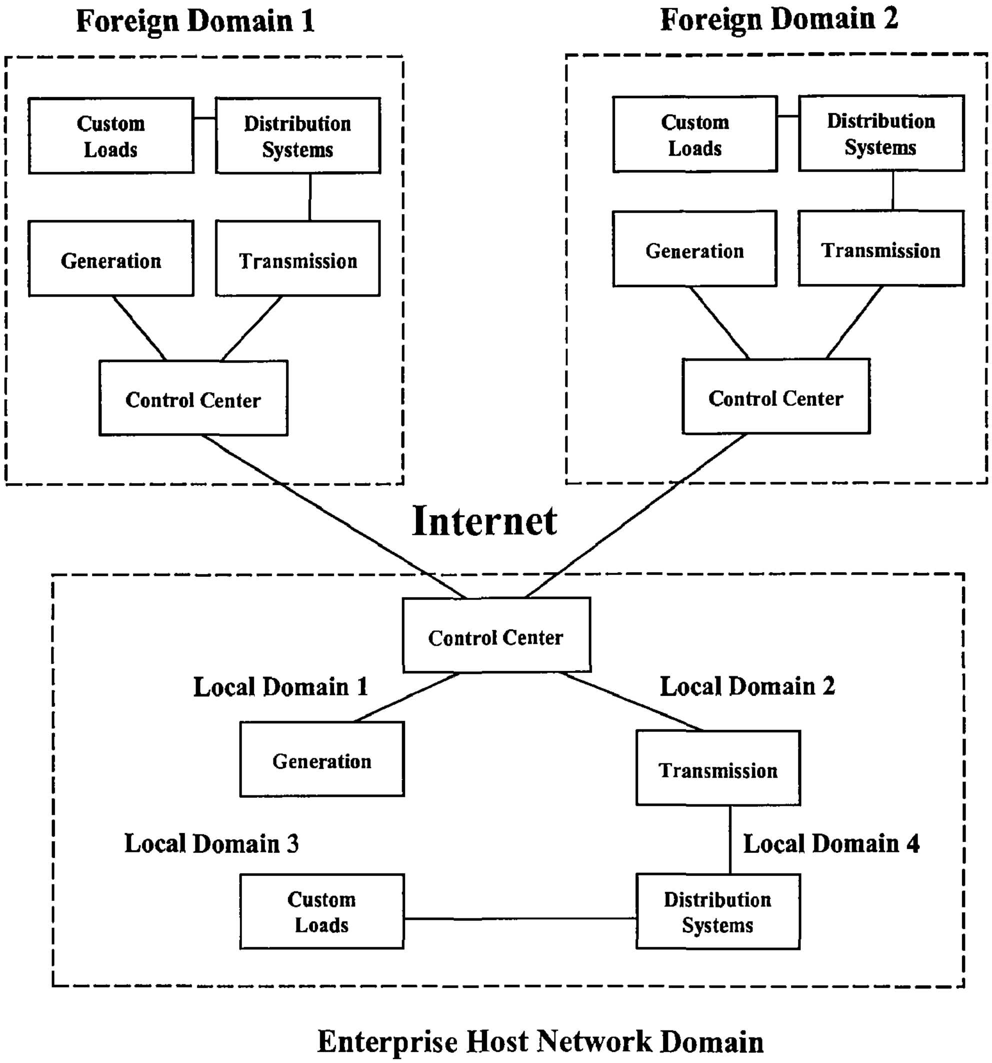


Figure 3.1: Security Management Architecture

3.2 Power-system Role-based Access Control (PRAC)

In this thesis research, a specific role-based access control model is developed for the power system computer networks. This model is named the Power-system Role-based Access Control (PRAC). The significant part of the PRAC model that differs from the conventional ones is: (1) PRAC defines the relationships between network roles in a local domain and those of interconnected local domains. (2) PRAC extends the access control from one power enterprise domain to foreign enterprise domains.

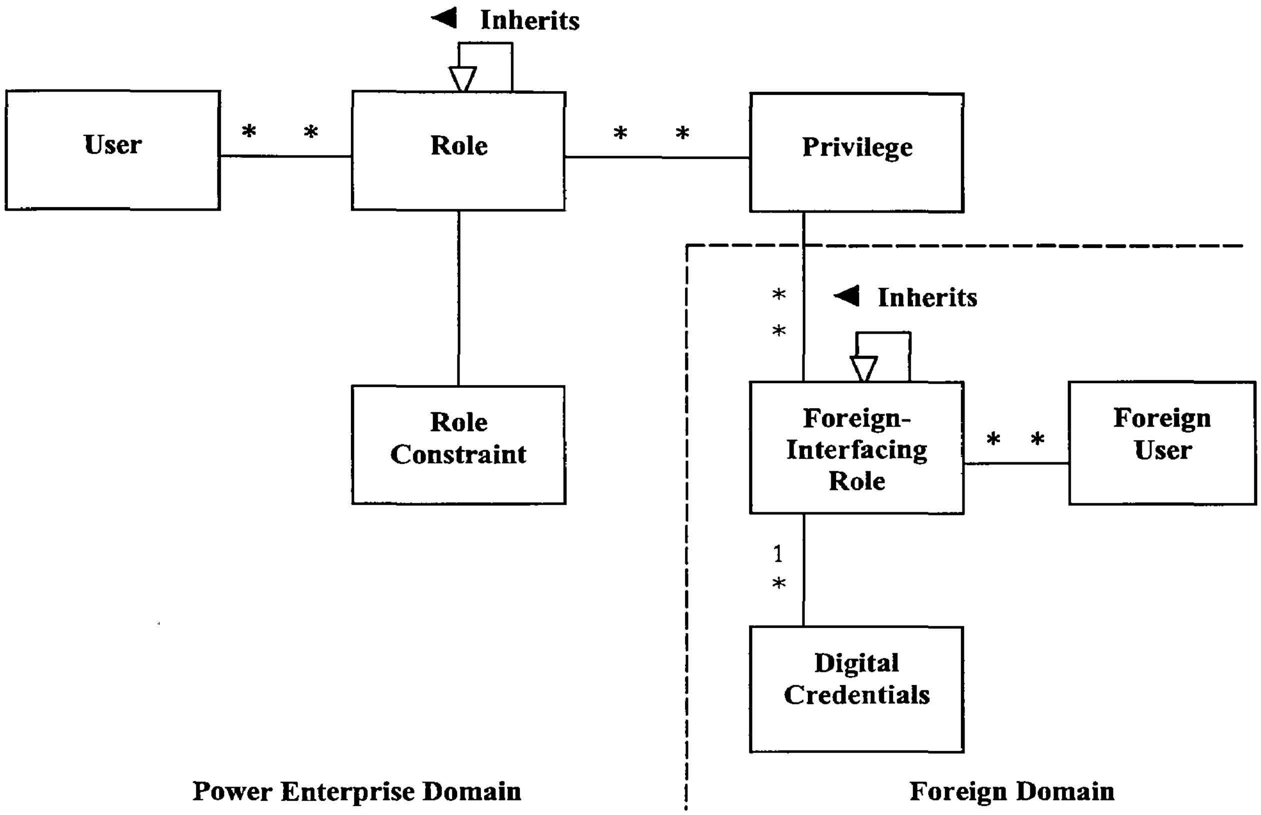


Figure 3.2: PRAC Model for Power System Computer Networks

Figure 3.2 shows the diagram of the PRAC model that consists of two parts: one part for a host domain, and one part for foreign domains. The designs of these two parts are discussed in the following.

3.2.1 PRAC Model for Host Domain

In the PRAC model for the enterprise computer network domain, a role is defined as a collection of privileges of network access that can be executed by the authorized users of certain job positions in the enterprise. A role can take on a number of privileges according to its functions and authorities. A user can be assigned with a number of roles, and a role can be assigned to multiple users.

In general, a set of roles can be assigned to a particular user according to the user's responsibility and authority in the power enterprise. A privilege in the PRAC model is an access mode that can be exercised on objects, such as monitoring power system performance, trading electricity, operating substation equipments, etc.

3.2.1.1 Role Hierarchy of Host Domain

With the PRAC architecture created in this thesis research, the network control center is designed to have the authority and facilities to structure the enterprise domain role hierarchy. This design allows the control center computer network to have full capability to structure the network access according to the real power system environment as well as capable of adapting to the changes in the environment. The enterprise hierarchy can be comprised of multiple local-domain role hierarchies. A role hierarchy is established to represent inheritance of authority, responsibility, and privilege among the roles in the power enterprises. A simple role hierarchy is described as follows:

If role r_i points to role r_j (i.e. $r_i \rightarrow r_j$), then r_i inherits all privileges of $r_{j.}$ The role r_i is called a direct parent role of r_j, and r_j is called a direct child role of r_i. The inheritance relationship is transitive. For example, if $r_i \rightarrow r_j$ and $r_j \rightarrow r_{k,}$ then $r_i \rightarrow r_{k.}$ In this example, the role r_i is an indirect parent role of r_k, and r_k is an indirect child role of $r_{i.}$ Both direct and indirect parent roles are simply called the parent roles, and both direct and indirect child roles are called the child roles.

In an enterprise computer network, a role in a local domain may take on some tasks belonging to other local domains. Figure 3.3 shows an example of role hierarchies in local domain 1 and 2, where role C in local domain 1 is assigned to an access role G in local domain 2. Role C is then called an extended parent role of G, and G is called an extended child role of C.

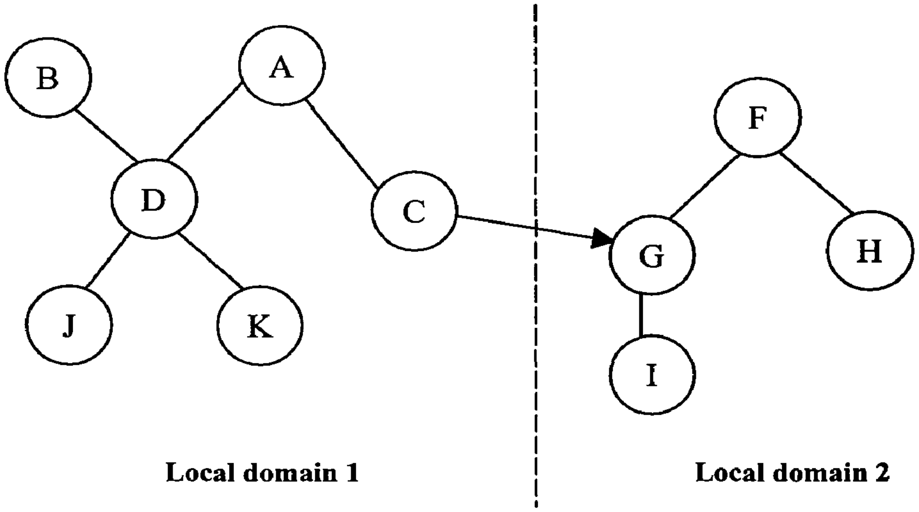


Figure 3.3: Role Hierarchy of Enterprise Domain

3.2.1.2 Role Constraints

The network controller in the PRAC model is responsible to evaluate constraints to the role operations defined with required qualifications. The PRAC model is designed to efficiently handle different types of role constraints. The role constraints [26] in general can be grouped into three types: cardinality constraint, separation-of-duty constraint and prerequisite constraint.

The cardinality constraints handled by the PRAC-based computer network include that a role may be allowed to have a limited number of users, a user may be allowed to execute a limited number of roles, etc.

The separation-of-duty constraints [27] are managed by the PRAC-based network. This management is to enforce conflict-of-interest prevention policies established by the control center for the access control. Conflict of interest arises as a result of the simultaneous assignment of two mutual exclusive roles to the same user. For example, a role of network monitor and a role of network operator cannot be assigned to the same user in order to avoid the conflict of interest in power enterprise systems.

The prerequisite constraints for specific network accesses are checked by the network controller in the PRAC-based network. The controller will ensure that a user can perform a prerequisite operation if and only if the user is already a member of prerequisite role. For example, a role with a privilege to initiate a tripping command to the substation bus-tie breaker is a role that has a prerequisite constraint. The constraint is determined by the control center such as the role must be a substation operator with a specific training and experience. A user can execute this role if and only if the user already has a role of substation operators. Moreover, there exist many other role constraints, for instance time constraint that defines how long the role can be activated.

3.2.2 PRAC Model for Foreign Domain

The PRAC model designed in this thesis research extends the capability of the conventional role-based access control model to cover the foreign electricity enterprise computer network domains. In the PRAC model, privileges can be given to users from the foreign domains. These users are called the foreign domain users. In order to deal with the access control of users from the foreign domains, it is required to establish a foreign-user network access policy to verify the trust relationship between the users of host domain and those of foreign domains.

Digital credentials can be used to manage the trust establishment efficiently [28-30]. The PRAC model designed in this thesis uses the digital credential to verify the network access request to the electricity power system computer networks. This credential verification is particularly important for maintaining the network security when the network is open to the foreign users. Digital credentials are the online counterparts of paper credentials that people use in their daily lives. There are multiple subject properties and their values for each type of credential. For example, the digital credential of a substation circuit breaker operator has several typical subject properties including the user's profession, specific trainings, specific experiences, etc. The control center network administrator (a software) automatically assigns a digital value to each subject property such as a profession engineer is represented with 9912345, training 1 with 456789, training 2 with 456788, etc.

Trust establishment using the digital credential is applicable whenever a foreign user requests to engage in a sensitive transaction without sufficient pre-established trust, and such a request involves essentially every aspect of the e-commerce in the power systems. An enterprise's network access policy may allow foreign users to access to certain data files but such access may limit to the authorized users.

In the PRAC model, a foreign-interfacing role in the host computer network domain can be viewed as a collection of privileges that can be performed by a foreign user who holds the required digital credentials. In the PRAC-based computer network, the control centre defines a security policy for the foreign-interfacing role that it can only be authorized to foreign computer network domain users who holds the required credential.

The security policy for digital credentials is one part of the XML-based PRAC security policy that will be discussed in the next section.

3.3 XML-based PRAC Security Policy System

In this thesis research, an XML-based PRAC security policy system is established for the electricity power system enterprise domain and for connections with the foreign computer network domains. The XML (eXtensible Markup Language) specification was established by the World Wide Web Consortium Standard Generalized Markup Language Working Group [31]. An advantage of using XML is that, as a meta-language, XML can effectively define a precise PRAC security policy system that can be extended or modified easily.

XML is employed for the syntactic representation of the PRAC security policy system. The XML representation is comprised of two parts: 1) Basic Elements, and 2) Relationships of Elements. The first part defines the elements of the PRAC model including role, privileges, constraints, and credentials. The second part defines the relationships among the elements of the PRAC model. There are four assignment sections to implement the relationships:

- Role hierarchy assignment for role-role relationships
- Privilege assignment for role-privilege relationships
- Constraint assignment for role-constraint relationships

- Credential assignment for role-credential relationships

3.3.1 Examples of XML-based Policy System

The following gives four examples of XML-based policy.

Example 1: Syntax for Elements and Relationships of Elements

In the XML-based PRAC security policy specification of the host compute network domain, the syntax of basic elements such as role, privilege, and constraint is defined. For example, role is represented by:

```
<! –Role definition-- >
    <ROLE ID=role-id ></ ROLE>
<! –Role definition-- >
```

In this format, a new XML tag of type ROLE with a required ID attribute value role-id is specified.

The Syntax of element relationship for the host computer network domain is specified in Part 2 of Appendix A. A privilege assignment that assigns privilege to a role is represented by:

```
<! -- Privilege assignment definition-- >
   <PRIV-ASSIGN  ROLE = ti  PRIVILEGE=mi.></PRIV-ASSIGN>
<! -- Privilege assignment definition-- >
```

This syntax defines a new XML tag of type PRIV-ASSIGN with ROLE, and PRIVILEGE attributes in which role *ti* has a privilege *mi*. The detailed syntax of all elements and their relationships are given in Appendix A.

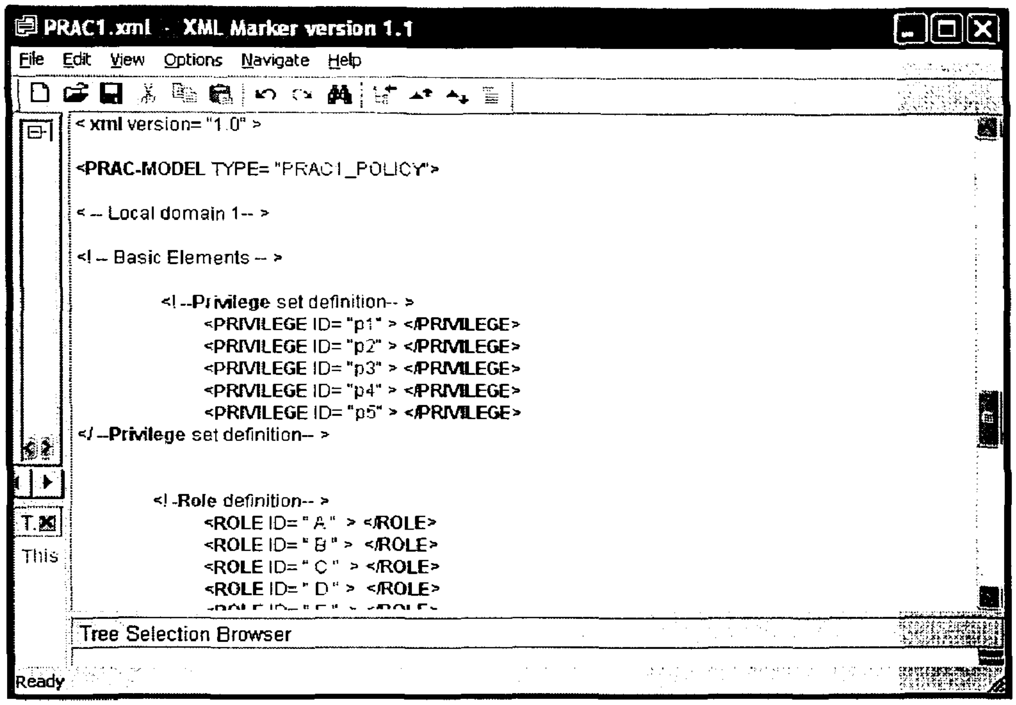
Example 2: Privilege Assignment

An example of privilege assignment in the host domain is shown below. Five privileges p1, p2, p3, p4 and p5 are included and each privilege is assigned to a specific role.

```
<! -- Basic Elements -- >
    <! --Privilege set definition-- >
        <PRIVILEGE ID= "p1" > </PRIVILEGE>
        <PRIVILEGE ID= "p2" > </PRIVILEGE>
        <PRIVILEGE ID= "p3" > </PRIVILEGE>
        <PRIVILEGE ID= "p4" > </PRIVILEGE>
        <PRIVILEGE ID= "p5" > </PRIVILEGE>
    </ --Privilege set definition-- >
<! -- Basic Elements -- >


< !-- Relationships of Elements -- >
    <! -- Privilege assignment definition-- >
        <PRIV-ASSIGN  ROLE= "A" PRIVILEGE=  "p1"></PRIV-ASSIGN>
         <PRIV-ASSIGN  ROLE= "B" PRIVILEGE=  "p2, p3"></PRIV-ASSIGN>
         <PRIV-ASSIGN  ROLE= "D" PRIVILEGE=  "p4, p5"></PRIV-ASSIGN>
    </ -- Privilege assignment definition-- >
< !-- Relationships of Elements -- >
```

The detailed description of the PRAC policy for the host domain is given in Appendix B. An interface of the XML-based policy system is shown in Figure 3.4.



PRAC1.xml - XML Marker version 1.1
File Edit View Options Navigate Help
T.
This
Tree Selection Browser
Ready

```
< xml version= "1.0" >

<PRAC-MODEL TYPE= "PRAC1_POLICY">

< -- Local domain 1-- >

<! -- Basic Elements -- >

          <! --Privilege set definition-- >
               <PRIVILEGE ID= "p1" > </PRIVILEGE>
               <PRIVILEGE ID= "p2" > </PRIVILEGE>
               <PRIVILEGE ID= "p3" > </PRIVILEGE>
               <PRIVILEGE ID= "p4" > </PRIVILEGE>
               <PRIVILEGE ID= "p5" > </PRIVILEGE>
</ --Privilege set definition-- >


         <! -Role definition-- >
               <ROLE ID= " A "  > </ROLE>
               <ROLE ID= " B " >  </ROLE>
               <ROLE ID= " C "  > </ROLE>
               <ROLE ID= " D " >  </ROLE>
```

Figure 3.4: Interface of XML-based Security Policy for Enterprise Host Domain

Example 3: Syntax for Foreign-Interfacing Role

The Syntax of the foreign-interfacing role is represented by,

<! – Foreign-Interfacing Role definition-- >
    <FOREIGN-Interfacing ROLE ID=*role-id*></FOREIGN-Interfacing ROLE>
<! – Foreign-Interfacing Role definition-- >

This syntax defines an XML tag Foreign-Interfacing ROLE and an attribute ID of which the value is role-id.

XML specification also defines the syntax of element relationships for the foreign domain with foreign-interfacing role hierarchy for foreign interfacing role-role relationships, privilege assignment for foreign interfacing role-privilege relationships, and credential assignment for foreign interfacing role-credential relationships. The detailed syntax for all elements and their relationships of foreign domain are given in Appendix C.

An example, such as foreign-interfacing role hierarchy, is represented as follows:

```
<! –Foreign-Interfacing Role hierarchy definition-- >
        <INHERITES FROM = ri  To = rj></INHERITES>
<! –Foreign-Interfacing Role hierarchy definition-- >
```

In this format, foreign-interfacing role is represented as a set of INHERITES elements, each of which associates a foreign-interfacing role with its direct child role.

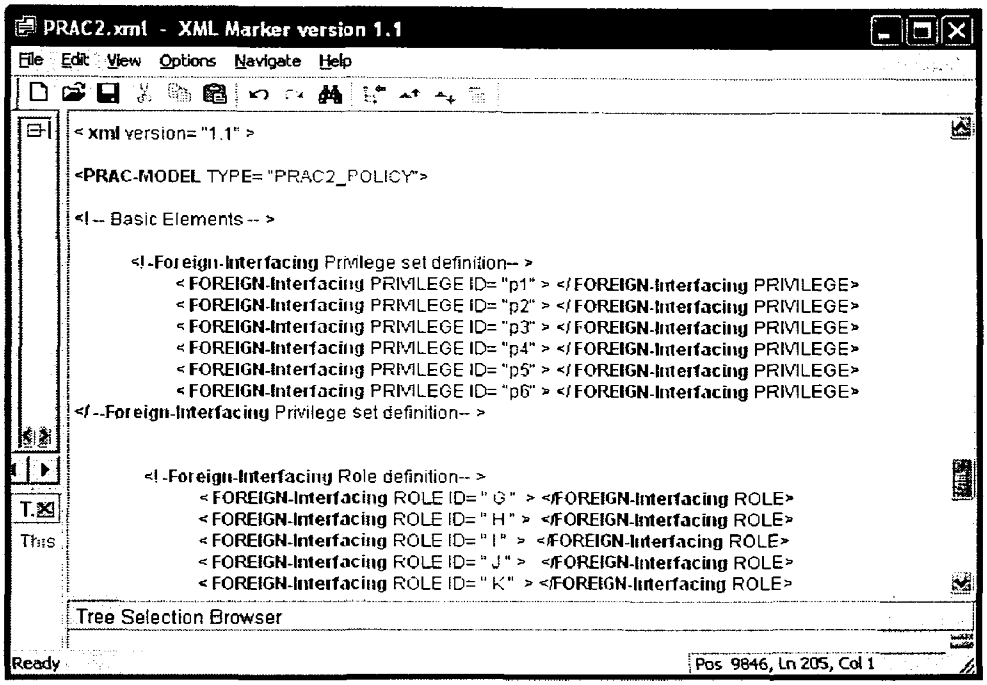
The syntax of foreign-interfacing role hierarchy defines a new XML tag of type INHERITES with a required FROM (foreign-interfacing role ri) and TO (foreign-interfacing role rj) attribute values which indicate foreign role ri is a direct parent foreign-interfacing role of rj.

Example 4: Foreign-Interfacing Role Assignment

An example of foreign interfacing role assignment in the foreign domain is shown below where foreign interfacing role G is a direct parent of H, foreign interfacing role H is a direct parent of I, etc.

```
<! –Foreign-interfacing role hierarchy definition-- >
        <INHERITS FROM = “G”  To  “H” ></INHERITS>
        <INHERITS FROM = “H”  To  “I”></INHERITS>
        <INHERITS FROM = “J”   To  “I” > </INHERITS>
        <INHERITS FROM = “K”   To  “H”></INHERITS>
</ --Foreign-interfacing role hierarchy definition-- >
```

The detailed description of XML-based security policy system for the foreign domain is given in Appendix D and the interface of the security policy is shown in figure 3.5.



PRAC2.xml - XML Marker version 1.1
File Edit View Options Navigate Help
T.
This
Tree Selection Browser
Ready
Pos 9846, Ln 205, Col 1

```
< xml version= "1.1" >

<PRAC-MODEL TYPE= "PRAC2_POLICY">

<! -- Basic Elements -- >

        <! -Foreign-Interfacing Privilege set definition-- >
              < FOREIGN-Interfacing PRIVILEGE ID= "p1" > </ FOREIGN-Interfacing PRIVILEGE>
              < FOREIGN-Interfacing PRIVILEGE ID= "p2" > </ FOREIGN-Interfacing PRIVILEGE>
              < FOREIGN-Interfacing PRIVILEGE ID= "p3" > </ FOREIGN-Interfacing PRIVILEGE>
              < FOREIGN-Interfacing PRIVILEGE ID= "p4" > </ FOREIGN-Interfacing PRIVILEGE>
              < FOREIGN-Interfacing PRIVILEGE ID= "p5" > </ FOREIGN-Interfacing PRIVILEGE>
              < FOREIGN-Interfacing PRIVILEGE ID= "p6" > </ FOREIGN-Interfacing PRIVILEGE>
</ --Foreign-Interfacing Privilege set definition-- >


          <! -Foreign-Interfacing Role definition-- >
                 < FOREIGN-Interfacing ROLE ID= " G "  > </FOREIGN-Interfacing ROLE>
                 < FOREIGN-Interfacing ROLE ID= " H " >  </FOREIGN-Interfacing ROLE>
                 < FOREIGN-Interfacing ROLE ID= " I "  >  </FOREIGN-Interfacing ROLE>
                 < FOREIGN-Interfacing ROLE ID= " J " >   </FOREIGN-Interfacing ROLE>
                 < FOREIGN-Interfacing ROLE ID= " K"  > </FOREIGN-Interfacing ROLE>
```

Figure 3.5: A Security Policy Interface of Appendix D

3.3.2 Administration of XML-based PRAC Security Policy System

The control center administrates the XML-based PRAC policy system for the host computer network domain. The network administrator of the control center defines the PRAC policy for the entire system as well as the policies for all local domains. There are four local domains in a typical power system computer network: generation domain, transmission domain, distribution domain, and customer load domain.

Each local domain loads its own XML-based PRAC policy from the control center. This allows the control center to administrate the PRAC policy for each local domain and to enforce consistence in policy administration. The PRAC policy has the same XML structure but different role hierarchies for different local domains. An example of the XML-based security policies loaded from the control center to Local Domain 1, 2, 3, and 4 is shown in Figure 3.6.

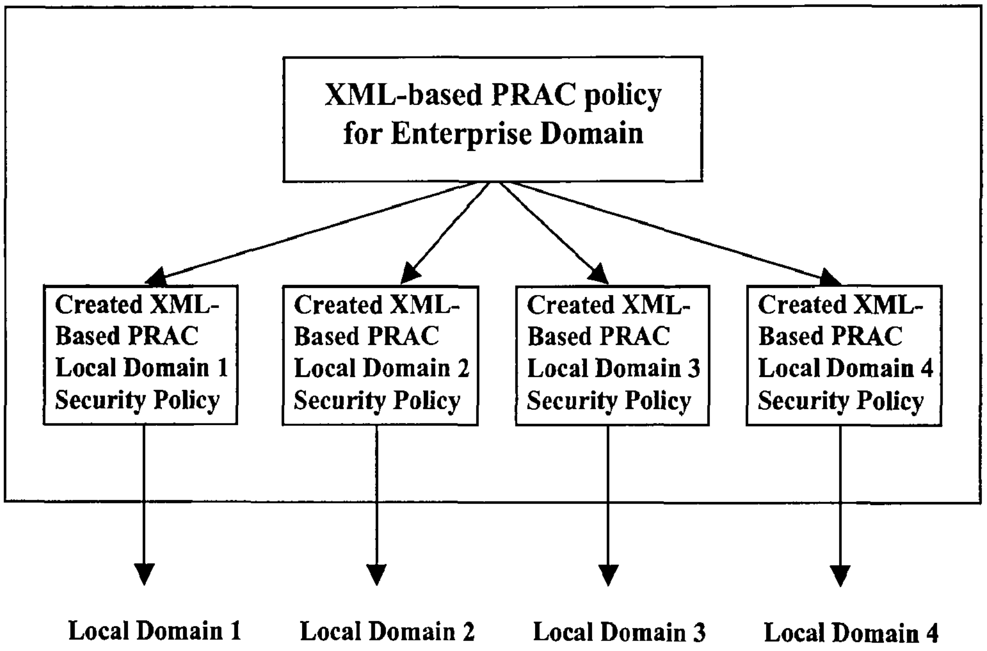

XML-based PRAC policy for Enterprise Domain
Created XML-Based PRAC Local Domain 1 Security Policy
Created XML-Based PRAC Local Domain 2 Security Policy
Created XML-Based PRAC Local Domain 3 Security Policy
Created XML-Based PRAC Local Domain 4 Security Policy
Local Domain 1
Local Domain 2
Local Domain 3
Local Domain 4

Figure 3.6: XML-based Policy Loaded From Control Center

3.4 Privilege Assignment Procedure for PRAC Model

This thesis has designed a procedure for assigning the privileges to the roles in the PRAC model for the power system applications. Three study cases shown in Figure 3.7 are considered in the following.

Case 1: User Access to its Own Local Domain

User X logs on Local Domain 1, the generation domain shown in Figure 3.7. User X is assigned with a role based on X's job functions and responsibilities.

The implementation of the PRAC model for this case can be divided into three steps:

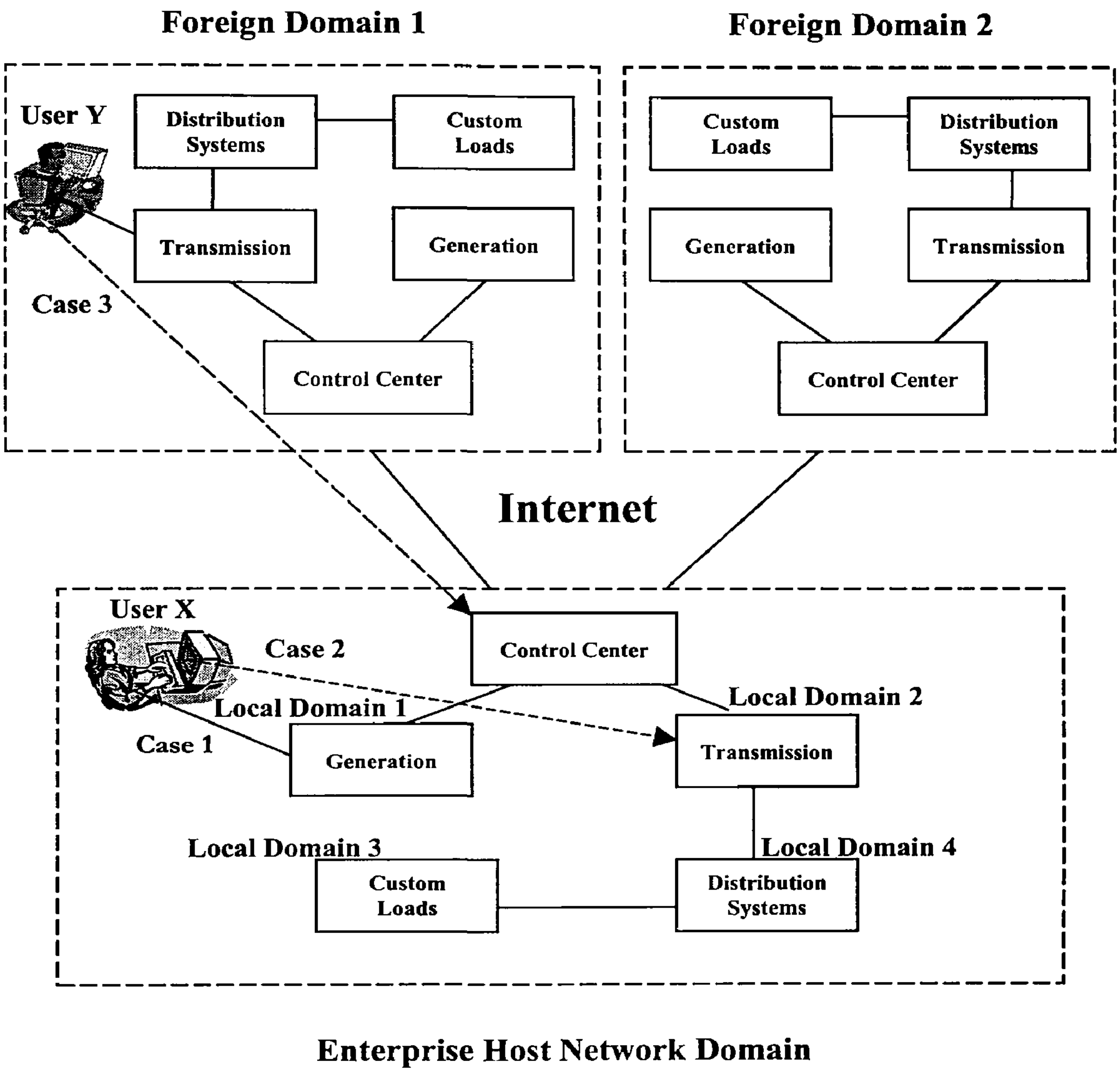


Figure 3.7: Procedure of the XML-based PRAC Security Policy

Step 1: User X requests for a privilege of operation such as generating power into transmission system, increasing the generator bus voltage, etc.

Step 2: Local Domain 1 administrator determines whether to authorize the operation to the user X according to its local domain PRAC policy.

Step3: User X is allowed to implement those operations if the privilege is authorized.

Case 2: User Access to Another Local Domain

User X of Local Domain 1 logs on Local Domain 2 (the transmission domain shown in Figure 3.7). The implementation of the PRAC model for this case can be divided into four steps:

Step 1: User X requests a privilege for an operation in Local Domain 2 such as transmitting power from generating station to the specified load, tripping the breaker on the high voltage transmission system, etc.

Step 2: Local Domain 1 administrator sends the request to Local Domain 2 administrator after determining that user X holds an extended parent role of Local Domain 2.

Step 3: Local Domain 2 administrator determines whether to authorize the access to user X according to Local Domain 2 PRAC policy, in which the extended parent role that user X holds has the privilege of the requested access.

Step 4: User X is allowed to implement this access if the privilege is authorized.

Case 3: Foreign User Access to Host Domain

User Y from Foreign Domain 1, the transmission domain shown in Figure 3.7, requests for a privilege of an access in Host Domain, such as buying power from the transmission domain of the host, browsing the online electricity-trading database, etc. These requests are handled by the control center administrator in

Host Domain. The implementation of the PRAC model for this case can be divided into four steps:

Step 1: User Y of Foreign Domain 1 sends the request to the control center administrator of Host Domain.

Step 2: The control center administrator of Host Domain responds to User Y with a web page that guides Y to upload its digital credentials according to the PRAC foreign security policy of Host Domain.

Step 3: The control center administrator of Host Domain validates uploaded digital credentials of User Y and determines whether to authorize the privilege of request operation to User Y.

Step 4: User Y is allowed to implement these operations if the privilege is authorized.

The above has presented the method for managing policy using XML. The motivation of this method is to simplify PRAC policy administration for power system computer networks. Unlike most existing implementations for other applications, with this method, the authorization is independently defined and is separated from policy representation and from implementation mechanisms.

3.5 Concluding Remarks

This chapter has presented a security management architecture that was specially designed for power system computer networks. This chapter has illustrated that this architecture is capable of handling an electricity enterprise host domain and multiple foreign domains. This chapter has presented the PRAC model, a specific role-based access control model developed for the power system computer networks. This chapter has shown that PRAC extends the capability of the conventional role-based access control models for handling multiple computer network domains. This chapter has established an XML-based PRAC security policy system for the electricity power system enterprise domain and for connections with the foreign computer network domains. Examples have been given to illustrate this policy system. Finally this chapter has presented a procedure for assigning the privileges to the roles in the PRAC model for the power system applications. Three study cases have been given to illustrate the procedure.

Chapter 4

Power-system Role-based kEy Management (PREM)

This chapter details a new key management designed in this thesis for electricity power system computer networks. This key management is the core of the network access control developed in Chapter 3. This new management is named as Power-system Role-based kEy Management (PREM) in this thesis. The PREM uses one-way hash functions and utilizes their advantages of computationally efficient and irreversibility. The irreversibility characteristic of one-way hash functions, selected by this thesis for PREM implementation, offers security in the role-based hierarchy structure such that the roles of lower levels are virtually impossible to derive the keys of the higher levels for unauthorized accesses.

This chapter presents the development of the PREM for the role hierarchy in a local network domain. There are four core components in the PREM. The first core component is the rules for key generation designed in this thesis research. This chapter illustrates that the rules will uniquely define the key for each role, as the same rules are used for both key generation and key derivation.

The second core component of the PREM is the key management designed in this thesis research for dynamic hierarchy. Under the open access policy, the hierarchy for the electricity power system computer networks with multiple domains subjects to dynamic changes that include frequently adding, deleting, and changing roles or objects, or modifying the relationships among roles as well as role and object. For whatever dynamic changes in the hierarchy, the key management of the PREM only needs to deal with the affected portion in the hierarchy. This chapter will show this significant advantage of the PREM in efficient key management, particularly for the power system computer networks with dynamic hierarchy due to the open access pressure.

The third core component of the PREM consists of two algorithms: one for the key generation and one for key modification. Both algorithms are based on the key generation rules to calculate the key for a role or for an object in a role hierarchy. The fourth core component of the PREM is the procedure for object assignments. The procedure is designed for the user to access encrypted objects such as encrypted files.

In this chapter, the PREM is extended from its prime design for one power system computer network local domain to the design that can handle multiple local domains. The extended PREM, developed in this thesis research, controls the access to the multiple local domains by developing two additional core components. First an architecture for the extended PREM is specially designed to manage the access for multiple local role hierarchy domains. Second, an object assignment protocol for multiple local domains is developed in this thesis research. The extended PREM decentralizes the key management for each local domain that is independently managed by its own local domain administrator. Any change of the role hierarchy structure in one local domain does not affect the keys of other local domains.

The following lists the sections in this chapter.

Section 4.1 presents the PREM for efficient single-domain key management. This section shows 4 core components in the PREM that include rules for key generation, key management for dynamic hierarchy, algorithms for key generation and key modification, and procedure of object access using keys.

Section 4.2 extends the PREM from one local domain to multiple local domains in the power system computer network. A key management architecture and an object assignment protocol are designed as the core components of the extended PREM.

Section 4.3 provides the concluding remarks of this chapter.

4.1 PREM for Single Local Domain

This section presents the development of the PREM for the role hierarchy in a power system computer network local domain. The PREM developed in this thesis research uses one-way hash functions. The two main advantages of using the one-way hash function [32] for implementation of the PREM are: one for computationally efficient and one for security of irreversibility. The irreversibility characteristic of the one-way hash function is utilized to offer high security in the role-based hierarchy structure such that the roles of lower levels are virtually not possible to derive the keys of the higher levels for unauthorized accesses.

For a given role hierarchy that includes roles and objects in a local domain, a set of one-way hash functions $\{H_1, H_2, \ldots, H_n\}$ are chosen in the PREM to generate the keys, where n is the maximum number of direct child roles that a role can directly access in the role hierarchy. These hash functions are public known. In a role hierarchy, a role is called a dead-end role if it has no direct parent roles. As an example shown in Figure 4.1, role A and D are dead-end roles. A key generated for a role is called as a role key, and for an object is called as an object key. The keys are generated by the local domain security administrator.

The following discusses the core components of the PREM that include the basic rules for key generation, the key management for dynamic hierarchy, the algorithms for key generation and key modification, and the procedure of object access using keys.

4.1.1 Rules for Key Generation

This section presents the first core component of the PREM that is the rules for key generation designed in this thesis research. Also this section illustrates that the rules will uniquely define the key for each role, as the same rules are used for both key generation and key derivation. The rules are given below:

Rule 1: For a dead-end role, the network security administrator assigns it an arbitrary key.

Rule 2: If a role has only one direct parent, then the key of this role will be generated by the hash function H_i (K), where *i* means that the role is the i^{th} direct child of its direct parent role (form left to right), and K is the key of its direct parent role.

Rule 3: If a role has more than one direct parents, then the key of this role will be generated by a combination of hash functions in such a way that the key equals to H_i (H_i (K_x), H_j (K_y), ...), where K_x , K_y , ... are the keys of its direct parent roles (from left to right), and *i*, *j*, ... means that the role is the *i*th (from left to right) direct child role of the parent role with key K_x , the *j*th (from left to right) direct child roe of the parent role with key K_y , etc.

Figure 4.1 shows an example illustrating that the above rules can uniquely define the key for each role. In Figure 4.1, since the maximum number of direct child roles for every role in the role hierarchy is two, two hash functions are chosen, such as H_1 and H_2. Suppose that key K1 is assigned to the dead-end role A and key K2 is assigned to the dead-end role D. The keys for all non-dead-end roles in the role hierarchy are generated as follows:

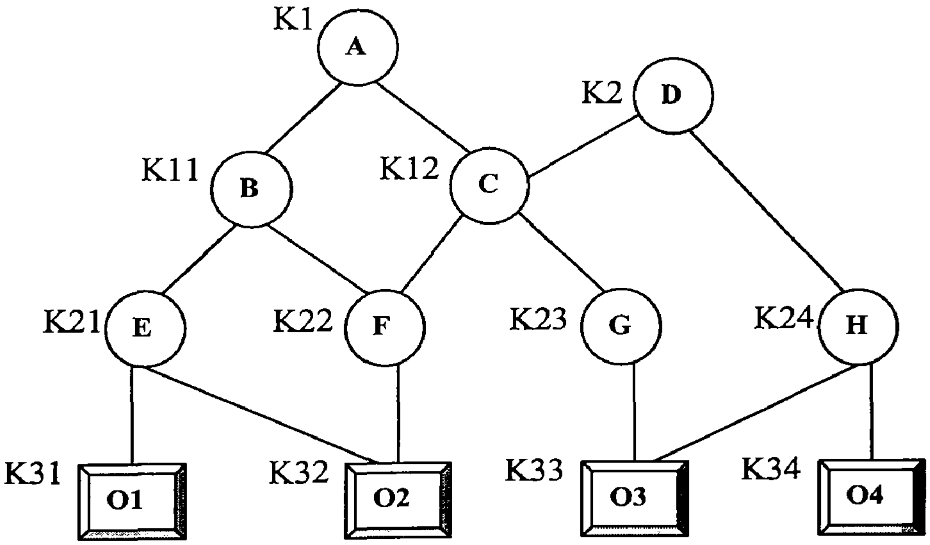


Figure 4.1: Derived Keys for Roles and Objects in a Role Hierarchy

For the roles that have only one direct parent roles, such as role B, the key of role B, K11, will be H_1 (K1). For the roles that have more than one direct parent roles, such as role C, the key of C, K12, will be H_2 (H_2 (K1)), H_1 (K2)). However, in order to derive the key of role C from role A or D, some parameters must be known. Role D must know the value of H_2 (K1), Role A must know the value of H_1 (K2). To be noted that only the roles that have more than one direct parent roles should have those parameters.

In a role hierarchy, a role may have privileges on one or more objects. In order to encrypt or decrypt the object for a role in a role hierarchy, the method that used to generate the key for the objects is solved in a way that is similar to the role keys. The difference for the object key generation method is that an object key value is same as the key value of a role if it is the only role to the object. In Figure 4.1, the key value of object O1 is same as the key value of role E and the key value of object O4 is same as the key of role H. If an object is directly accessed by more than one role, the object key generation method follows the key generation rules. For instance, the object key K32 is derived as H_2 (H_2 (K21), H_1 (K22)).

4.1.2 Key Management for Dynamic Hierarchy

This section presents the second core component of the PREM that is the key management designed in this thesis research for dynamic hierarchy. Under the open access policy, the hierarchy for the electricity power system computer networks with multiple domains subjects to dynamic changes that include frequently adding, deleting, and changing roles or objects, or modifying the relationships among roles as well as role and object. For whatever dynamic changes in the hierarchy, the key management of the PREM only deals with the affected portion in the hierarchy. This section will show this significant advantage of the PREM in efficient key management, particularly for the power system computer networks with dynamic hierarchy due to the open access pressure.

A) Adding a Role or an Object

If a role is added as a dead-end role, the local domain security administrator assigns a key to this new dead-end role and regenerates the keys of its direct and indirect child roles as well as the associated objects. For instance, in Figure 4.2, if a new dead-end role N is added, the role keys of C, D, F, G and H as well as object keys of O2, O3 and O4 are regenerated.

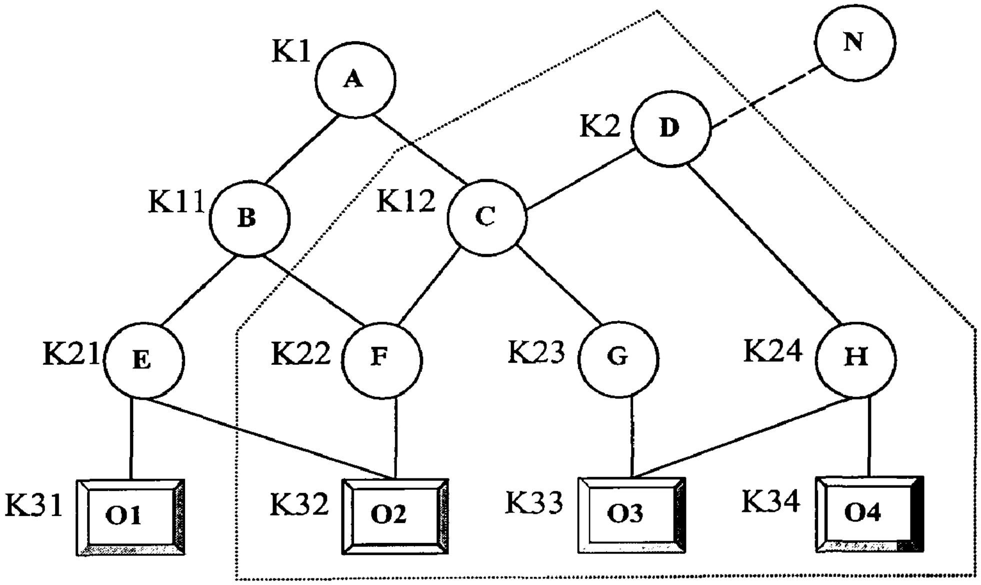


Figure 4.2: Adding a Dead-end Role

If an added role is not a dead-end role, the local domain security administrator derives the key of this added role from its direct parent roles according to the key generation rules, and then regenerates the keys of its direct and indirect child roles. For example, in Figure 4.3, if the new role N is added as the direct child role of A and direct parent role of C and H, the role key of N is then generated from role A. The role keys of C, F, G and H as well as object keys of O2, O3 and O4 are also regenerated.

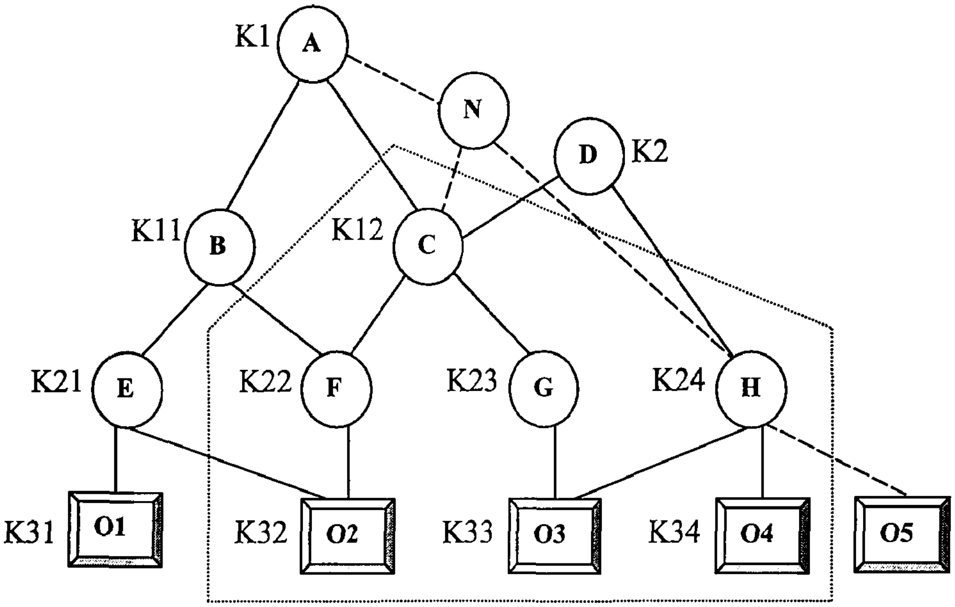


Figure 4.3: Adding a Non-dead-end Role or an Object

If an object is added to the role hierarchy, the new object key is generated according to the key generation rules. For example, an object O5 is added to role H. Similar to add a non-dead-end role, the object key of O5 is generated from role H.

B) Deleting a Dead-end Role or an Object

If a dead-end role R is deleted and one of its direct child roles S becomes the new dead-end role in the role hierarchy after the deletion, then the original key of S does not need to change. If a dead-end role R is deleted and one of its direct child roles S just looses one of its direct parents without becoming a new dead-end role, then the keys of S and S's direct and indirect child role are regenerated.

For example, in Figure 4.4, the dead-end role A is deleted and role B becomes a new dead-end role after the deletion of A. The key of role B does not need to change. Only the role keys of C and its direct and indirect child roles are regenerated. To delete an object has no effects on the roles or other objects. Therefore, all of the keys are kept same as before.

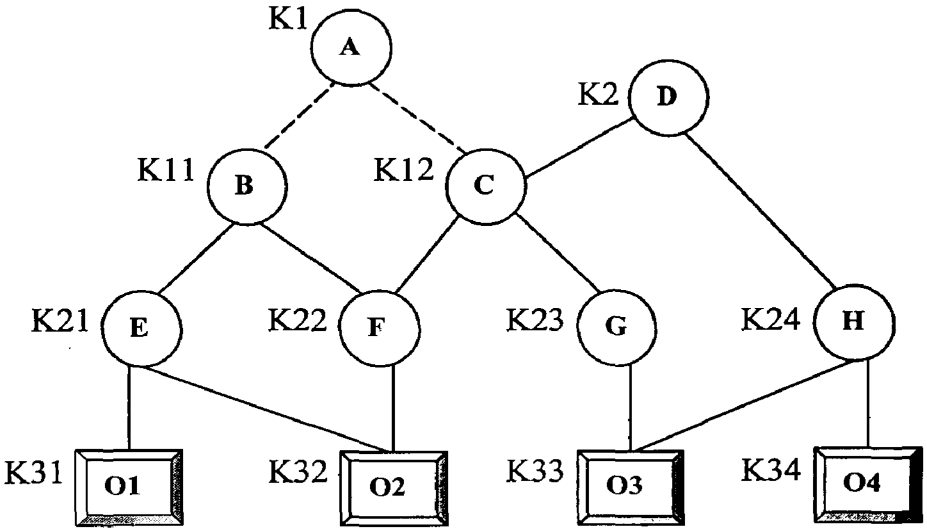


Figure 4.4: Deleting a Dead-end Role

C) Changing Relationships Among Roles and Objects

Suppose that role R is originally a direct parent role of S, if role S becomes a dead-end role after deleting the relationship between R and S, the key of S does not need to be changed. Otherwise, the keys of S and its direct and indirect child roles need to be regenerated. For example, if the relationship between role A and B, in Figure 4.5, is deleted and role B becomes a new dead-end role, then the key of B does not need to be changed. If the relationship between role A and C is deleted, the role keys of C, F and G as well as the object keys of O2 and O3 need to be regenerated.

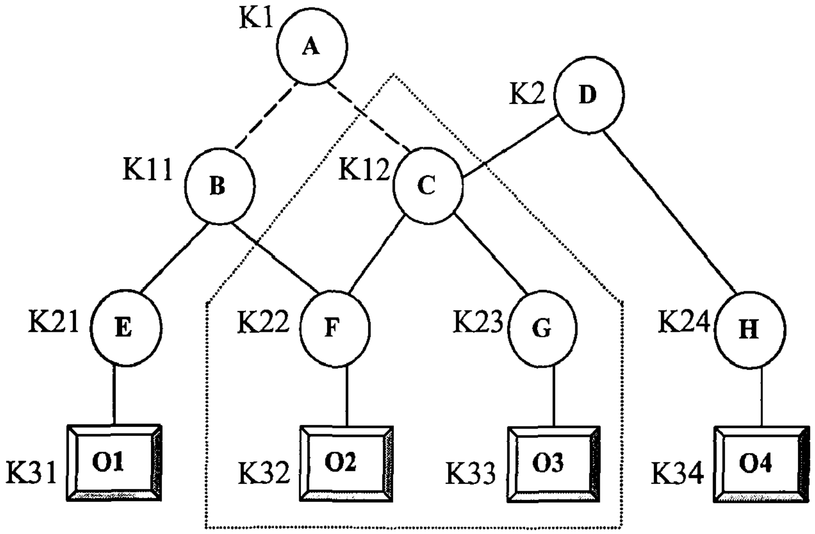


Figure 4.5: Deleting a Relationship Among Roles

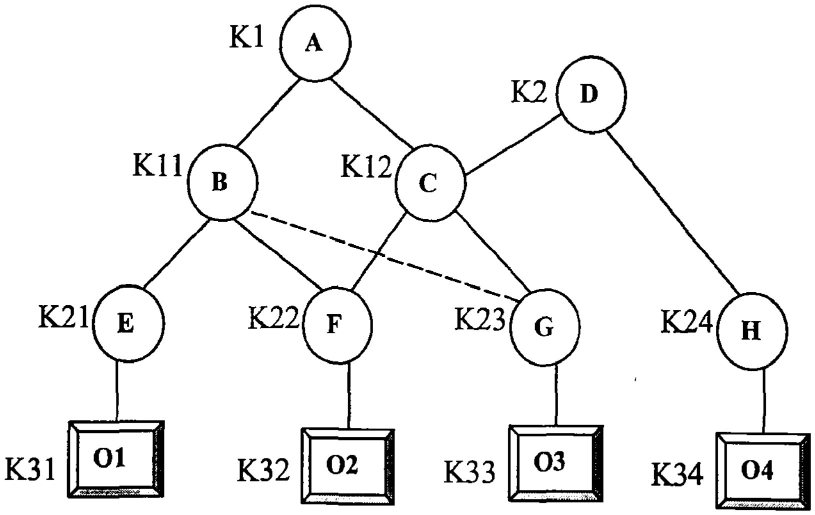


Figure 4.6: Adding a Relationship Among Roles

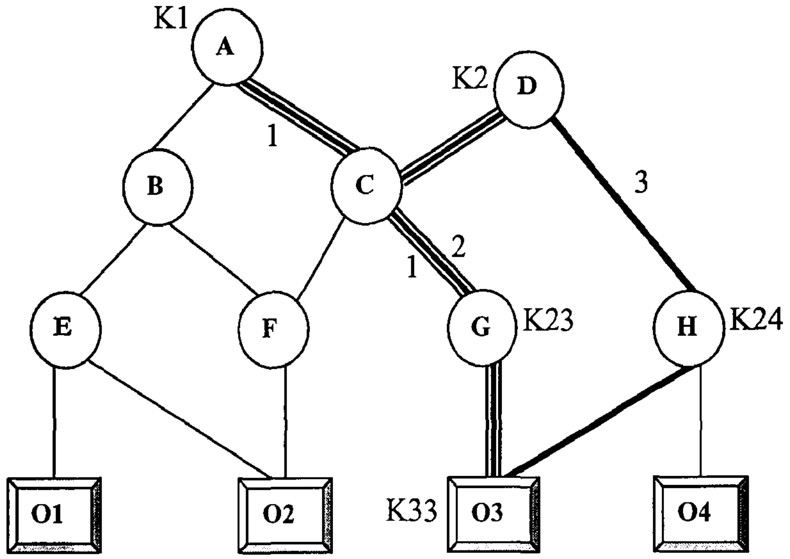
If a new relationship is added between two roles R and S, and R becomes a direct parent role of S, then the object keys of S and its child roles need to be regenerated. For example, if a new relationship between role B and G, shown in Figure 4.6, is added, then the key of role G and the key of object O3 need to be regenerated.

4.1.3 Algorithms for Key Generation and Modification

This section presents the third core component of the PREM that consists of two algorithms: one for the key generation and one for key modification. Both algorithms are based on the key generation rules to calculate the key for a role or for an object in a role hierarchy. An example of a role hierarchy is shown in Figure 4.7.

There are one or more than one paths from a dead-end role to the role or object whose key needs to be derived. To reduce the calculation time of generating a key value, the algorithm searches for the least-expense path from each dead-end role to the role or object. According to the key generation rules, the time used to calculate a key value of a role or an object depends on the number of hash function and the time used by each hash function.

Assume that all hash functions cost same time, then the execution time to calculate a key value depends on the number of hash functions in each path. For example, in Figure 4.7, to calculate the key value of object O3, we have three different paths: (1) D-H-O3 (2) D-C-G-O3 and (3) A-C-G-O3. The number of hash functions used in path (1) is 3+4+8=15, the number of hash functions in path (2) is 3+4+8=15, the number of hush functions in path (3) is 1+8=9. Thus, path (3) should be the least-expense path that is shown in Figure 4.7. If there are more than one least-expense path, the algorithm will pick up any of them. Finally, the algorithm calculates the key of the role or object based on the key generation rules introduced in section 4.1.1.



$K12 = H_2(H_2(K1), H_1(K2))$ $K23 = H_2(K12) = H_2(H_2(H_2(K1), H_1(K2)))$

$K24 = H_2(K2)$ $K33 = H_1(H_1(K23), H_1(K24))$

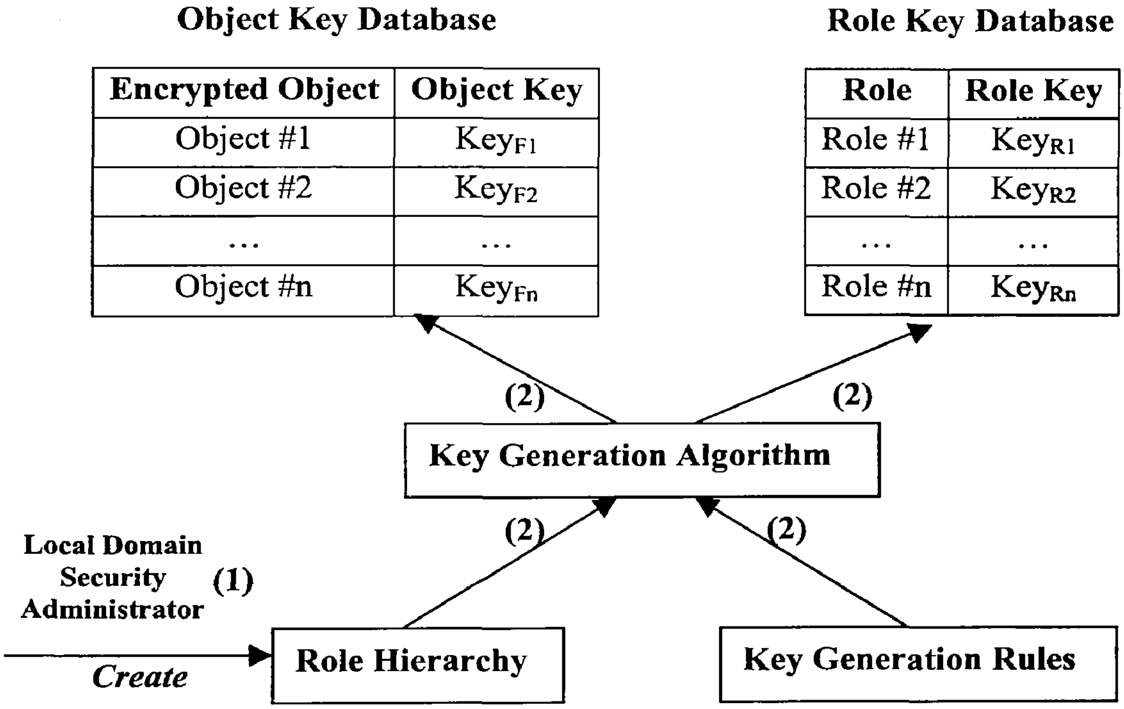
Figure 4.7: An Example of Key Generation

The key generation algorithm is shown as follows:

A) Algorithm 1: Key Generation Algorithm

```
/* For a given role or object R in a role hierarchy, calculate the key value of R
{
    For each dead-end role in the role hierarchy {
    Search for the least-expense path between the role or object R and the dead-end role
    }
    Calculate the key value of the role or object according to the key generation rules
    defined in section 4.1.1.
    Output the key value of the role or object R.
}
```

The local domain security administrator is in charge of generating key values for roles and objects. The generated role keys and roles will be stored in role key database. The object key and the encrypted objects will be stored in the object key database. The general procedure of the key generation includes two steps shown in Figure 4.8.



Object Key Database

Encrypted Object	Object Key
Object #1	Key$_{F1}$
Object #2	Key$_{F2}$
...	...
Object #n	Key$_{Fn}$

Role Key Database

Role	Role Key
Role #1	Key$_{R1}$
Role #2	Key$_{R2}$
...	...
Role #n	Key$_{Rn}$

Figure 4.8: Key Generation and Object Encryption in a Local Domain

Step 1 The local domain security administrator creates its local domain role hierarchy.

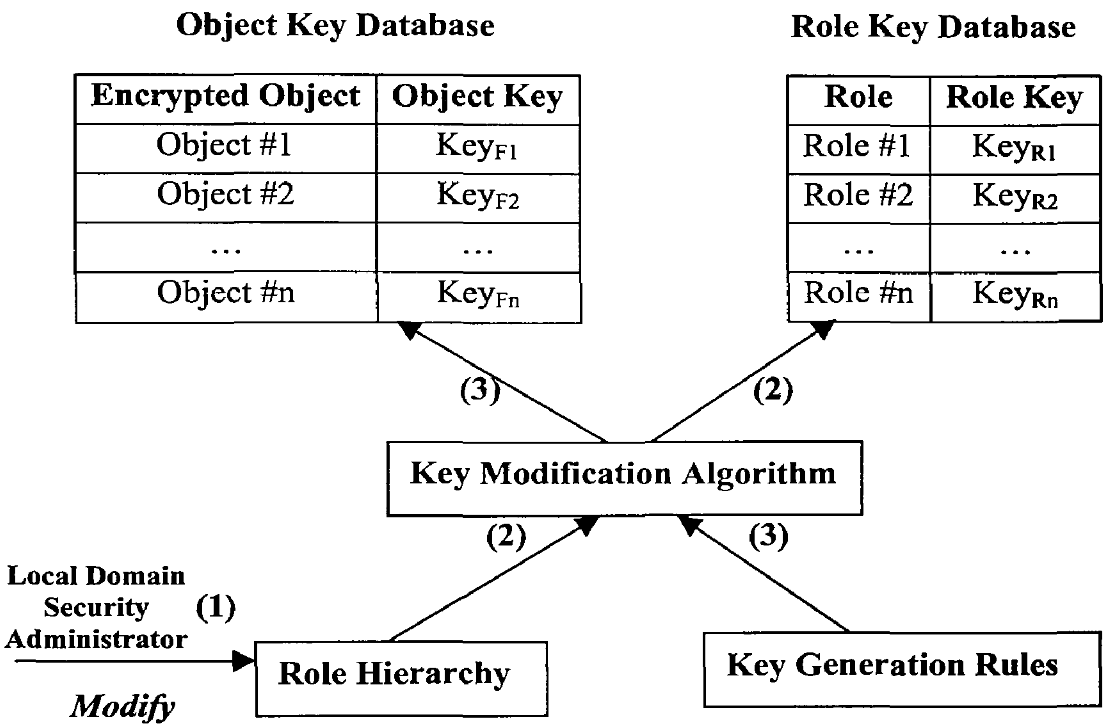
Step 2 The role hierarchy is searched using a searching algorithm such as breath-first search or depth-first search. For each role being searched, the key generation algorithm is called to generate the key value of the role. Role and its key value are saved in the role key database. For each object being searched, the key generation algorithm is called to generate the key value of the object, then the generated key is used to encrypt the object. The encrypted object and key value of the object are saved in the object key database.

Besides generating key values for roles and objects in a local domain, the local domain security administrator is also in charge of dynamically updating key values for roles and objects in the role hierarchy. A key modification algorithm is presented.

When the role hierarchy is modified (detailed has been discussed in section 4.1.2), the key modification algorithm searches for all the roles and objects that are affected. For each affected role, the key generation algorithm will be invoked to update its key value in the role key database. For each affected object, the key generation algorithm will be invoked to calculate the new key for the object, decrypt the encrypted object with the old key and re-encrypt the affected object with the new key and update the object and its key value in the object key database.

B) Algorithm 2: Key Modification Algorithm

```
Key Modification Algorithm ()
{
    Search the role hierarchy and for each role being affected by a modification {
        Call key generation algorithm and calculate the new key value for the role
        Update the key value in the role key database
        }
        Search the role hierarchy and for each object being affected by a modification {
                Call key generation algorithm and calculate the new key value for the object
                Decrypt the object with old key and re-encrypt the object
                Update the key value and object in the object key database
        }
}
```



Encrypted Object	Object Key
Object #1	Key_{F1}
Object #2	Key_{F2}
...	...
Object #n	Key_{Fn}

Role	Role Key
Role #1	Key_{R1}
Role #2	Key_{R2}
...	...
Role #n	Key_{Rn}

(3)
(2)
Key Modification Algorithm
(2)
(3)
Local Domain Security Administrator
(1)
Modify
Role Hierarchy
Key Generation Rules

Figure 4.9: The General Procedure for Key Modification in a Local Domain

The general procedure of the key modification is comprised of the following three steps and shown in Figure 4.9.

Step 1 The local domain security manager modifies the role hierarchy.

Step 2 The key generation algorithm is invoked to generate new keys for roles that are affected by the modifications. The key value of each affected role is updated and saved in the role key database.

Step 3 The key generation algorithm is invoked to generate new keys for objects that are affected by the modifications. Each affected object is decrypted first with the old key and re-encrypted by the new generated object key. The key value and the object are updated in the object key database.

4.1.4 Procedures for Object Assignments

This section presents the fourth core component of the PREM that is the procedure for object assignments. If a local domain user requests to access encrypted objects such as encrypted files, the user's membership of role should be verified by a database called the database of roles and users. The database of roles and users includes user's username, password, member of role and a symmetric key. The symmetric key is shared with his/her local domain security administrator and used for secure communication between the local domain user and the local domain security administrator. The general procedure of the object assignment method is comprised of the following five steps and shown in Figure 4.10.

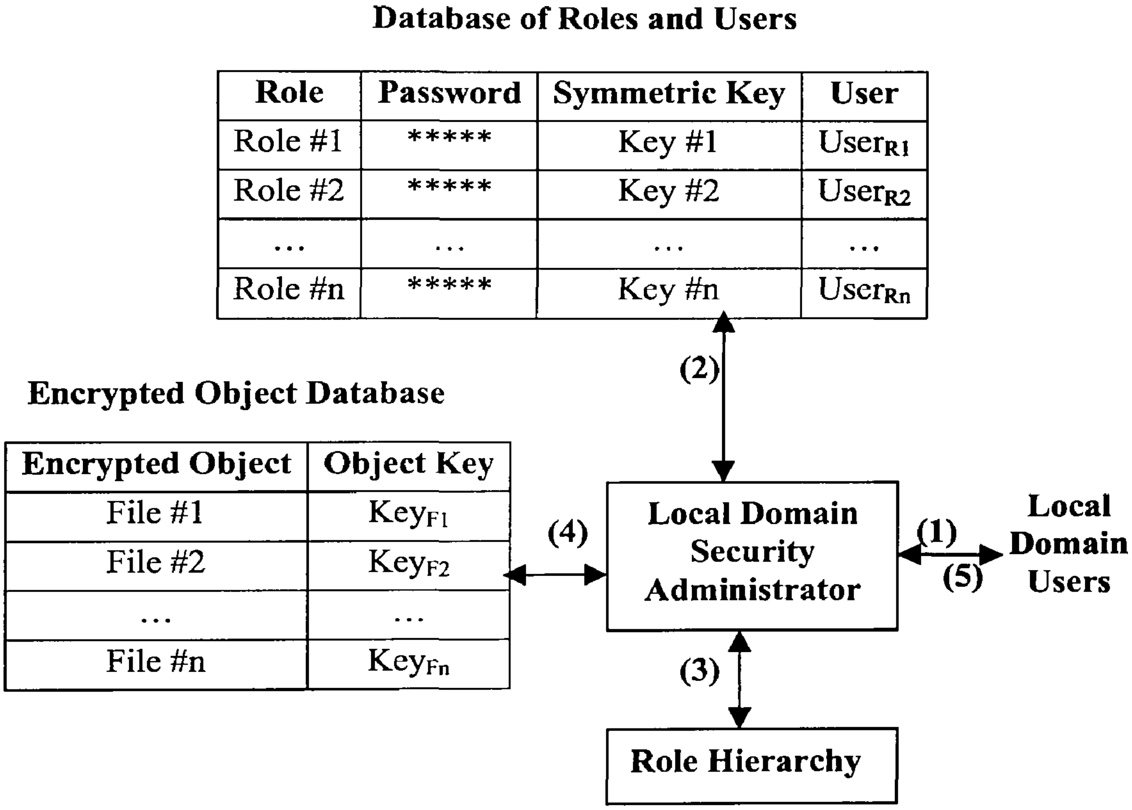
Step 1: A local domain user S requests to access an encrypted object such as an encrypted file T. His/her username and password are required to be submitted to verify his/her role membership.

Step 2: The local domain security administrator verifies the user's role membership by checking the database of roles and users.

Step 3: The local domain security administrator checks the role hierarchy to determine if the role S holds can access encrypted file T.

Step 4: If S can access file T, the local domain security administrator checks the object key database and decrypt the encrypted file T using its object key. Then, the decrypted file is encrypted again with the symmetric key shared by the local domain security administrator and user S. The symmetric key is stored in the database of roles and users.

Step 5: This encrypted file is sent to the local domain user S. User S can decrypt it using the symmetric key.



Database of Roles and Users

Role	Password	Symmetric Key	User
Role #1	*****	Key #1	User$_{R1}$
Role #2	*****	Key #2	User$_{R2}$
...	...	...	...
Role #n	*****	Key #n	User$_{Rn}$

Encrypted Object Database

Encrypted Object	Object Key
File #1	Key$_{F1}$
File #2	Key$_{F2}$
...	...
File #n	Key$_{Fn}$

Figure 4.10: Procedure of the Object Assignment Method

4.2 PREM Extended for Multiple Local Domains

This section shows that the PREM is extended from one power system computer network local domain to the multiple local domains. The extended PREM, developed in this thesis research, controls the access to the multiple local domains by developing two additional core components. First, an architecture for the extended PREM is specially designed to manage the access for multiple local role hierarchy domains. Second, an object assignment protocol for multiple local domains is developed in this thesis research. The extended PREM decentralizes the key management for each local domain that is independently managed by its own local domain administrator. Any change of the role hierarchy structure in one local domain does not affect the keys of other local domains.

In an enterprise domain, there are hundreds of privileges and thousands of users located in different local domains. The enterprise domain can be divided into multiple local domains; each local domain has a role hierarchy that is administrated by its local domain security administrator.

Figure 4.11 shows an example of an enterprise domain that includes role hierarchies of local domains 1 and local domain 2. In the role hierarchy of local domain 1, role D is a direct parent role of C and H, role C is a direct parent role of F and G and role D is an indirect parent role of F and G. Similarly, in local domain 2, role I is a direct parent role of role J and K, role J is a direct parent role of role L and M, and role I is an indirect parent role of L and M.

A role in one local domain may perform some tasks from other local domains and therefore needs to have privileges to access to other local domains. For instance, if role C in local domain 1 needs to have privileges of role M in local domain 2, role C has to be an extended parent role of M in order to perform M's privileges. Similarly, Role I in local domain 2 is an extended parent role of D in local domain 1 and D is an extended child role of I.

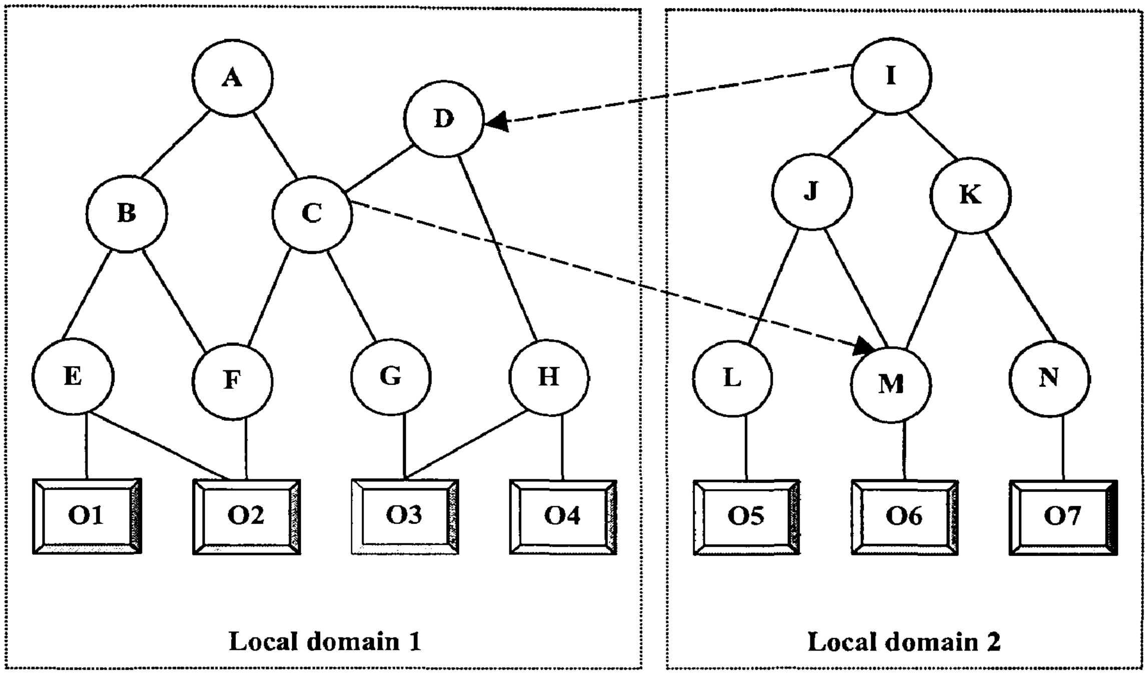


Figure 4.11: An Example of Role Hierarchies in Multiple Local Domains

The key management in multiple local domains is related to the inter-relationships of the roles in different local domains. Each local domain has its own role hierarchy that is administrated by its local domain security administrator. If a single key generation algorithm is applied for multiple local domains, any modification in a local domain could affect the key value of roles in other local domains. For instance, if the key value of role C in local domain 1 is changed, the keys of F and G in local domain 1 will be changed. Also, the key values of role M in local domain 2 will be changed. When multiple local domains inter-connected with each other, the key management work will be a nightmare for local domain security administrators.

To deal with this problem, a security management architecture of PRAC model for multiple local domains is designed.

4.2.1 Key Management Architecture for Multiple Local Domains

The key management architecture is comprised of multiple local domains. Each local domain includes a local domain security administrator, an object key database, a role key database and the clients. Figure 4.12 gives an example of the key management architecture for the multiple local domains such as local domain 1 and local domain 2.

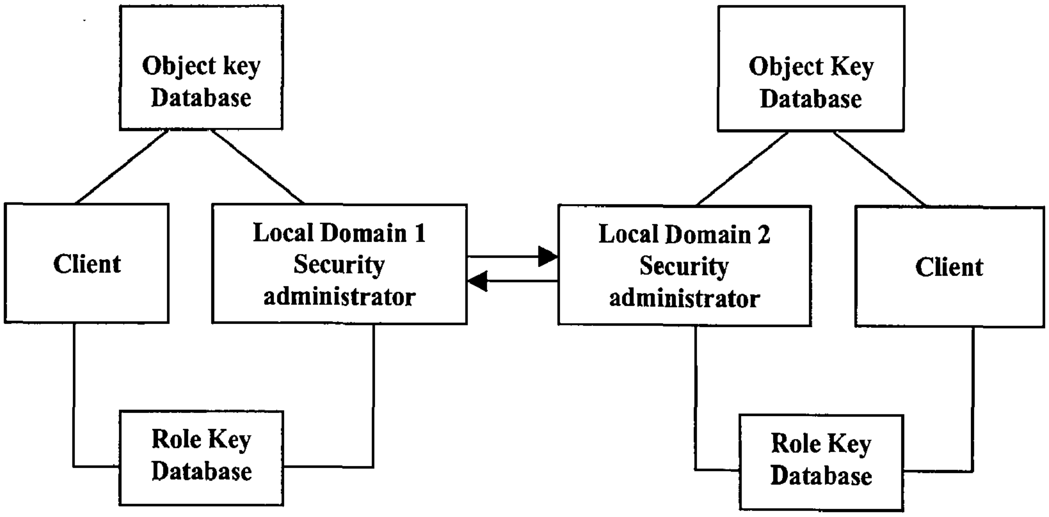


Figure 4.12: Key Management Architecture for Multiple Local Domains

The main functions of each element of the architecture are described as follows:

- The local domain security administrator is in charge of authorizing roles and objects to the users in its local domain as well as to the users of other local domains that requires to access the objects of this domain.
- The client accepts the application of accessing the objects from its local domain users and returns the authorized objects back to the users.
- Role keys are stored in the local domain role key database. Encrypted objects and their corresponding object keys are stored in the object key database.

4.2.2 Object Assignment Protocol for Multiple Local Domains

Based on the key management architecture presented above, an object assignment protocol for multiple local domains is developed. A user in one local domain must hold an extended parent role in order to access an object in other local domains. A negotiation must be implemented between different local domain security administrators in order to establish the access between an extended parent role and an extended child role.

Suppose that in local domain 1, a user who is a member of a role R wants to access an encrypted object T in local domain 2, the user has to first give the request to the local domain 1 security administrator.

Step 1: The administrator in local domain 1 first authenticates the user and its membership of the role R that the user holds, then give the request to the administrator in local domain 2.

Step 2: The administrator in local domain 2 determines whether the object T can be accessed through an extended child role of R.

Step 3: If the access is established between role R and its extended child role in local domain 2, the administrator in local domain 2 encrypts the object and sends it to the administrator of local domain 1.

Step 4: The domain 1 administrator decrypts the object using the key that is shared with the domain 2 administrator, and then send it to the user.

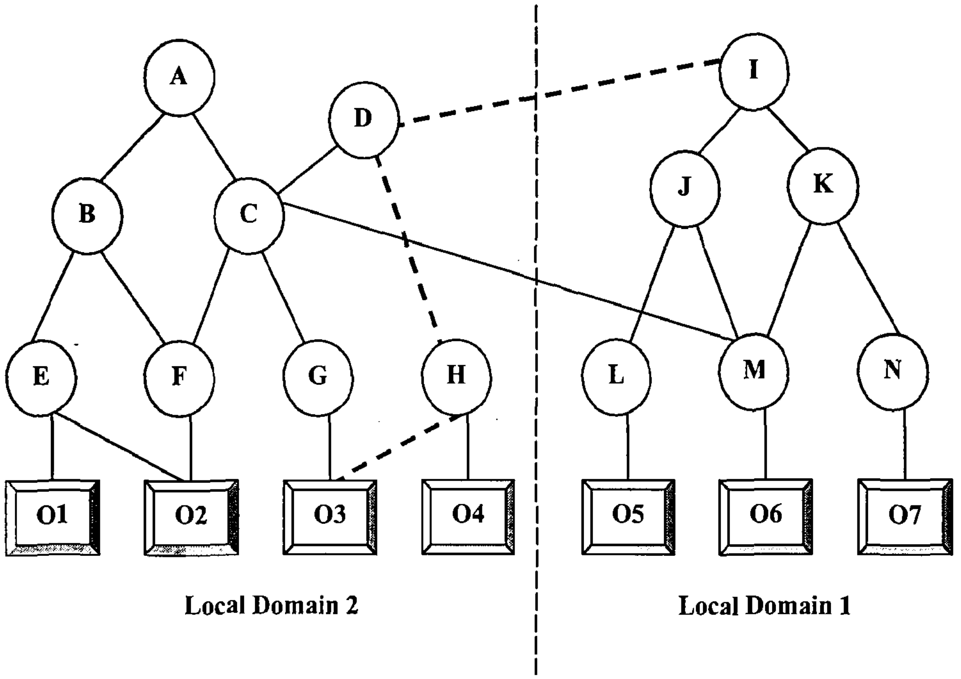


Figure 4.13: Inter Relationship of Role Hierarchies in Multiple Local Domains

For example, in Figure 4.13, if a user with role I in local domain 1 requests to access object O3 in local domain 2, the user must first apply the request to the administrator of local domain 1. After two domain administrators negotiate, an extended parent-child relationship is established between role I and role D. The administrator of local domain 1 authenticates the user and its membership of the role, and then sends the user's request to the administrator in local domain 2. The administrator of local domain 2 determines whether object O3 can be access by the extended child role D of Role I in local domain 2. Once determined, the administrator of local domain 2 encrypts object O3 with the encryption key that is shared with the administrator of local domain 1, and sends it out to local domain 1 security administrator. The administrator of local domain 1 decrypts the object O3 and sends to the user of role I in local domain 1.

4.3 Concluding Remarks

This chapter has detailed a new key management named PREM in this thesis for electricity power system computer networks. This chapter has shown that PREM uses one-way hash functions and utilizes their advantages of computationally efficient and irreversibility. The irreversibility characteristic of one-way hash functions, selected by this thesis for PREM implementation, offers security in the role-based hierarchy structure such that the roles of lower levels are virtually impossible to derive the keys of the higher levels for unauthorized accesses. This chapter has presented the development of the four core components in the PREM. The first core component is the rules for key generation designed in this thesis research. This chapter has illustrated that the rules uniquely defines the key for each role. The second core component is the key management for dynamic hierarchy. For whatever dynamic changes in the hierarchy, the key management of the PREM only needs to deal with the affected portion in the hierarchy. This chapter has shown this significant advantage of the PREM in efficient key management, particularly for the power system computer networks with dynamic hierarchy due to open access pressure.

This chapter has shown that the PREM is extended from its prime design for one power system computer network local domain to the design that can handle multiple local domains. The extended PREM, developed in this thesis research, controls the access to the multiple local domains by developing two additional core components. An architecture for the extended PREM is specially designed to manage the access for multiple local role hierarchy domains. An object assignment protocol for multiple local domains is developed in this thesis research. The extended PREM decentralizes the key management for each local domain that is independently managed by its own local domain administrator. Any change of the role hierarchy structure in one local domain does not affect the keys of other local domains.

Chapter 5

Platform for Comprehensive Assessment of PREM

This chapter presents a platform for the comprehensive assessment of PREM, an efficient and high security key management developed in Chapter 4, for the role-based access control of power system computer networks. For applications in real power systems, it is absolutely important to fully assess the security and efficiency of any network access control method prior to its implementation. However in practice, it is very difficult particularly within the university environment, to thoroughly assess the network access control. As a groundwork for remedying this real-world difficulty, this chapter proposes a simple platform for comprehensive assessment of the PREM.

This chapter presents two stages in the simple platform for fast and economical, but not thorough and not precise, assessment of the PREM. The first stage for functionality assessment of PREM uses two typical cryptographic algorithms: one for one-way hash function, one for symmetric key encryption. The second stage in the platform for benchmark assessment of PREM is via the comparison using typical existing key management method. Three study cases are given in this chapter to illustrate the assessment of the two key management methods.

The following lists the sections in this chapter.

Section 5.1 presents the first stage of a simple platform for assessing the PREM with the typical cryptographic algorithms.

Section 5.2 presents the second stage of a simple platform for assessing the PREM via the comparison with typical existing key management method.

Session 5.3 provides the concluding remarks for this chapter.

5.1 PREM Assessment Platform with Typical Cryptographic Algorithms

This section presents the first stage of a simple platform for assessing the PREM using two typical cryptographic algorithms: one for one-way hash function, one for symmetric key encryption.

A) One-way Hash Function: MD5

The MD5 (Message Digest) is a one-way hash function, meaning that it takes a message and converts it into a fixed string of digits, also called a message digest. When using a one-way hash function, one can compare a calculated message digest against the message digest that is decrypted with a public key to verify that the message hasn't been tampered with. The performance of MD5 may vary wildly from 30 to 90 M bit per second [33], depending on the network computing facilities.

B) Symmetric Key Encryption: Triple DES

Triple DES is a symmetric cryptographic algorithm. Its performance has been tested on a PC running Linux with a Pentium III CPU of 860MHz and 256MB of RAM, with the encryption/decryption operations on a 10 MB file, and the encryption algorithms using a key size of 112 bits [34]. The CPU execution time for key generation is 2.9 seconds, for encryption is 12.5 seconds, and for decryption is 12.6 seconds.

5.2 PREM Assessment Platform via Typical Key Management

This section presents the second stage of a simple platform for assessing the PREM via the comparison with typical existing key management method. As a fact in all key management methods, a modification of the network hierarchy, or its role and an object, or the relationship of roles and objects in the hierarchy most likely will affect the key values of other roles or objects. Compared with the typical key management method proposed by Akl and Taylor discussed in Chapter 2, the PREM updates much less number of key values for the roles and objects when new roles are added to the role hierarchy. Suppose the performance comparison between the two key management methods is based on the same role hierarchy and the each object is the same size of 10 MB. In the following, three study cases are given to illustrate the assessment of the two key management methods.

Case Study 1: Generating Keys for Roles and Objects in a Role Hierarchy

To generate the keys for roles and objects in a role hierarchy, the total time used, T, is comprised of three parts including the time for generating the keys of roles, T_{keyR} , the time for generating the keys of objects, T_{keyO} , and the time for encrypting the objects, T_o. The total time to generate keys for a hierarchy is simply calculated by (5.1).

$$T = T_{keyR} + T_{keyO} + T_o \qquad (5.1)$$

Suppose that MD5 is chosen to generate the keys. The running time of one hash function is T_H , the number of hash functions used for each role j in the role hierarchy is N_j, and the number of roles in the role hierarchy is N_R , then the time for generating keys for all roles in the hierarchy is calculated in (5.2).

$$T_{keyR} = \sum_{j=1}^{N_R} T_H * N_j = T_H * \sum_{j=1}^{N_R} N_j \qquad (5.2)$$

Suppose that the number of objects in the role hierarchy is N_O , and the number of hash functions used for generating a key for each object b is N_b , then the time for generating keys for all objects in the hierarchy is calculated in (5.3).

$$T_{keyO} = \sum_{b=1}^{N_o} T_H * N_b = T_H * \sum_{b=1}^{N_o} N_b \qquad (5.3)$$

Suppose that the running time for encrypting an object is T_E , and the number of objects in the role hierarchy is N_O , then the time for encrypting all of objects in the hierarchy is calculated in (5.4). According to the calculations of (5.1), (5.2), (5.3) and (5.4), the total time used to generate keys of roles and objects in the hierarchy is calculated in (5.5).

$$T_o = T_E * N_O \qquad (5.4)$$

$$T = T_H * \sum_{j=1}^{N_R} N_j + T_H * \sum_{b=1}^{N_o} N_b + T_E * N_O \qquad (5.5)$$

When the PREM is compared with Akl and Taylor's key management method, for a given role hierarchy, both methods have the same values of T_H, N_R, N_O, and T_E . The PREM executes the hash functions more than Akl and Taylor's method, and therefore the values of N_j and N_H of the PREM are larger than those used in Akl and Taylor's method.

However, this time difference could be ignored because the hash functions such as MD5 uses much less execution time than the encryption time T_E and $T \approx T_E * N_O$. Therefore, the time used to generate the keys of roles and objects is almost same for both key management methods.

Case Study 2: Adding a Role in the Role Hierarchy

As key management discussed in Chapter 2, it is the fact that when a new role is added into a role hierarchy after all keys of roles and objects have been generated using Akl and Taylor method, the key value of new role may possibly be obtained from any of the other roles. This significantly degrades the security of the network access. To remedy this situation, a complete key generation procedure of Akl and Taylor method must be performed for all roles and objects. On the other hand, the PREM is capable of accommodating any networking changes due to adding new roles into the role hierarchy.

When a new role is added to the role hierarchy, the running time T_{ADD} is comprised of three parts including the time for updating the keys of the affected roles, T_{keyADD} , the time for decrypting the affected objects, T_{dADD} , and the time for re-encrypting the affected objects using updated new keys, $T_{eADD.}$ The total time to update the keys of the affected objects in a hierarchy, due to adding a new role, is simply calculated by (5.6).

$$T_{ADD} = T_{keyADD} + T_{dADD} + T_{eADD} \qquad (5.6)$$

Suppose that the running time of a hash function is T_H. The number of hash functions used for a role j is N_j . The number of the affected roles is N_r , when a role is added in the role hierarchy. The number of the affected object is N_f and the number of hash functions used for each object t is N_t , The time for generating all affected keys is calculated in (5.7).

$$T_{keyADD} = \sum_{j=1}^{N_r} T_H * N_j + \sum_{t=1}^{N_f} T_H * N_t = T_H * (\sum_{j=1}^{N_r} N_j + \sum_{t=1}^{N_f} N_t) \quad (5.7)$$

Suppose that the running time of decrypting an object is T_D, the time for encrypting an object is T_E , and the number of affected objects is N_f when a new role is added in the role hierarchy, then the time for decrypting the affected objects is calculated in (5.8). According to the calculations of (5.6), (5.7), (5.8) and (5.9), the total time used to generate keys of roles and objects in the hierarchy is calculated in (5.10).

$$T_{dADD} = T_D * N_f \quad (5.8)$$

$$T_{eADD} = T_E * N_f \quad (5.9)$$

$$\begin{aligned} T_{ADD} &= T_H * (\sum_{j=1}^{N_r} N_j + \sum_{t=1}^{N_f} N_t) + T_E * N_f + T_D * N_f \\ &= T_H * (\sum_{j=1}^{N_r} N_j + \sum_{t=1}^{N_f} N_t) + (T_E + T_D) * N_f \\ &\approx (T_E + T_D) * N_f \quad (5.10) \end{aligned}$$

Comparing the PREM with Akl and Taylor method, for a given role hierarchy, if a new role is added, the different time value of T_{ADD} for both key management methods is mainly depends on the time for encryption and decryption as well as the number of the affected objects since the execution time of hash function is much more smaller than the execution time of encryption or decryption. Therefore, the total execution time of updating the keys, due to adding a new role to the hierarchy, approximately equals total time of encryption and decryption the affected objects.

For example, in Figure 5.1, suppose each object is 10 MB and a new role O is inserted in the role hierarchy. Role O is a direct child role of A and a direct parent role of H. If using Akl and Taylor method to update the keys of the affected object, all objects in the role hierarchy need to be regenerated. Suppose that 3DES is chosen as the encryption and decryption algorithm to implement both key management methods. As 3DES assessed in Section 5.1, the execution time to encrypt a 10MB file needs 12.5 seconds, and to decrypt the same size file needs 12.6 seconds. The running time T_{ADD} using Akl and Taylor method is calculated as follows:

$$T_{ADD} \approx (T_E + T_D) * N_f = (12.5+12.6)*8=200.8 \text{ seconds}$$

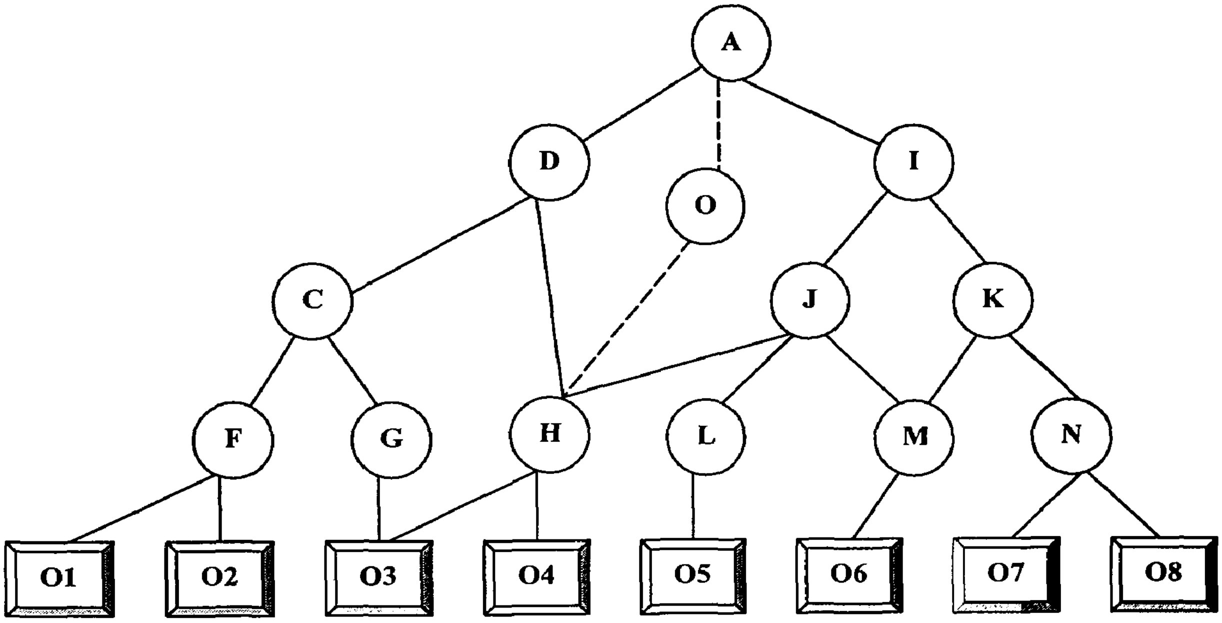


Figure 5.1: An Example of Role Hierarchies

If the PREM is used to update the keys of the affected objects, only the keys of objects O3 and O4 need to be regenerated. The key generation running time T_{ADD} of the PREM using 3DES is calculated as follows:

$$T_{ADD} \approx (T_E + T_D) * N_f = (12.5+12.6)*2=50.2 \text{ seconds}$$

Suppose that there are total 10000 objects in a role hierarchy. The running time of PREM and Akl and Taylor method is illustrated in Figure 5.2, in terms of number of affected objects. The total execution time difference indicates that the advantage of the PREM over Akl and Taylor method is distinct if the size of the objects in an enterprise is large and only small percentage objects are affected. Therefore, the PREM updates much less number of key values for the roles and objects when new roles are added to the role hierarchy.

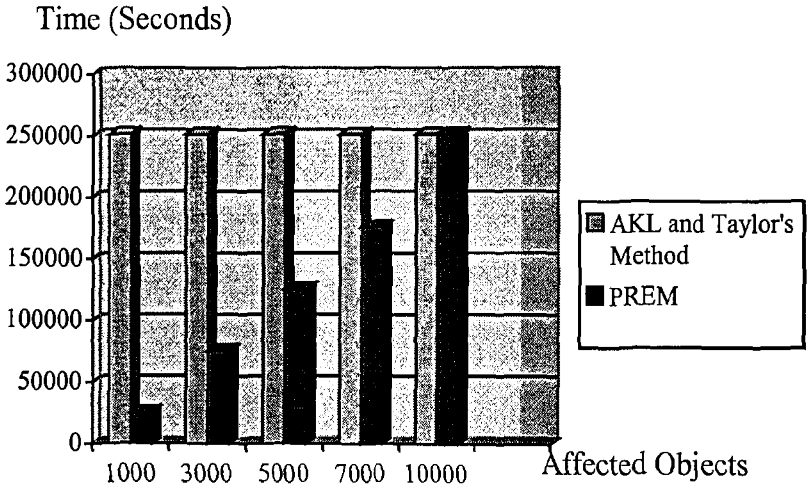


Figure 5.2: Comparing PREM with AKl and Taylor's Method

Case Study 3: Deleting Roles/Objects or Changing Relationships Between Roles

In a role hierarchy, if a role or an object is deleted or relationships between roles are changed, the same portion of objects are affected by using both key management method. Therefore, the running time is almost same under these situations.

5.3 Concluding Remarks

This chapter has presented a platform for the comprehensive assessment of PREM for the role-based access control of power system computer networks. Since in practice it is very difficult particularly within the university environment, to thoroughly assess the network access control, this chapter therefore has proposed a simple alternative that consists of two stages. The first stage for functionality assessment of PREM uses two typical cryptographic algorithms: one for one-way hash function, one for symmetric key encryption. The second stage in the platform for benchmark assessment of PREM is via the comparison using typical existing key management method. Three study cases have been given to illustrate the assessment.

Chapter 6

Conclusion

This thesis research has been successfully completed, with two developments: a new network access control, the PRAC, and a new key management, the PREM. The PRAC is an efficient Power-system Role-based Access Control that significantly increases the security of computer networks for power systems, and surmounts the challenges caused by typical security and reliability concerns due to current technological and political changes faced in the electricity power industry. The PREM is a secure Power-system Role-based kEy Management that is developed to support the efficient operation of the PRAC.

6.1 Major Research Work Completed

The following presents a summary of major tasks accomplished in this thesis research.

A. Exhaustive investigation has been conducted on the currently available techniques, models, and managements used for the computer network access control. The investigation has identified that existing techniques do not well satisfy the strict requirements for reliable power system operations that initiated this thesis research.

B. A security management architecture has been specially designed for power system computer networks. This architecture is capable of covering the network access control to the electricity enterprise host domain and its multiple foreign domains.

C. The PRAC model has been developed for the power system computer networks, which extends the capability of the conventional role-based access control models for handling multiple computer network domains. An XML-based PRAC security policy system has been established for the host domain and for connections with the foreign domains.

D. The PREM has been designed as an efficiency key management for the power system computer networks. This method uses one-way hash functions and utilizes their advantages of computationally efficient and irreversibility. Four core components in the PREM have been developed, that include rules for key generation, key management for dynamic hierarchy, algorithms for key generation and key modification, and procedure of object access using keys are discussed.

E. PREM has been designed for dynamic hierarchy access control. For whatever dynamic changes in the hierarchy, the key management of the PREM only needs to deal with the affected portion resulted from the changes.

F. PREM has been extended from its prime design for one power system computer network local domain to the design that can handle multiple local domains. The extended PREM controls the access to the multiple local domains by developing two additional core components: one is an architecture and one is an object assignment protocol. The architecture for the extended PREM is specially designed to manage the access for multiple local role hierarchy domains. The object assignment protocol is developed to guide and establish the access between multiple local domains.

G. This chapter has presented a platform for a comprehensive assessment of PREM. Since in real-world practice it is very difficult, particularly within the university environment, to thoroughly assess any network access control, this chapter therefore has proposed a simple alternative that consists of two stages. The first stage for functionality assessment of PREM uses two typical cryptographic algorithms: one for one-way hash function, one for symmetric key encryption. The second stage for benchmark assessment of PREM is via the comparison using typical existing key management method. Three study cases have been given to illustrate the assessment.

6.2 Major Research Contributions

1. ***PRAC, a specific role-based access control for single computer network,*** has been developed, based on findings of exhaustive investigations that were conducted in this thesis research for security improvement of power system computer network.

2. ***Extended PRAC, for multiple computer networks,*** has been designed for the access control to the host domain of the power system being considered and its interconnecting domains of neighboring power systems.

3. ***PREM, for single local computer network domain,*** has been developed for key management of dynamic hierarchy access control using one-way hash functions and utilizing their advantages of computationally efficient and irreversibility security.

4. ***Extended PREM, for multiple local computer network domains,*** has been designed for the host domain consisting a central domain for the power system control center network and several local domains for computer networks of generation systems, transmission systems, distribution systems, and customer loads.

5. ***Platform for comprehensive assessment of PREM*** has been established for fast and economical, but not thorough and not precise, assessment of the key management developed in this thesis research.

6.3 Future Work

The following presents recommended future work on the following areas:

- Extend the PRAC model and PREM methods to solve the authorization and security concerns over the Internet.

- Combine PRAC and PREM solution with the existing managing and monitoring security device for identifying cyber-based threats to the survivability of power substation control networks.

- Carry out the PRAC solution in the research of distributed database authorization and XML database authorization.

APPENDIX A

Specifications of XML-based PRAC Security Policy for Enterprise Host Domain

This thesis research employs XML for the syntactic representation of the PRAC security policy system. The XML representation is comprised of two parts: Basic Elements and Relationships of Elements. In this Appendix, the first part defines the elements of the PRAC model for the host domain including role, privileges and constraints. The second part defines the relationships among the elements of the PRAC model for the host domain.

Part 1: Defines the Basic Elements of PRAC Model for the Host Domain

- **Define Roles**

  Role is represented by:

  ```
  <! –Role definition-- >
      <ROLE ID=role-id ></ ROLE>
  ```

  The above syntax defines a new XML tag of type ROLE with a required ID attribute value role-id.

- **Define Privileges**

Privilege is represented by:

```
<! –Privilege set definition-- >
    <PRIVILEGE ID=privilege-id ></PRIVILEGE>
```

The above syntax defines a new XML tag of type PRIVILEGE with required ID attribute value privilege-id.

- **Define Constraints**

Constraint is represented by:

```
<! --Constraint set definition-- >
    <CONSTRAINT ID= constraint-id ></CONSTRAINT>
    <PARAMETER VALUE=parameter- values>< / PARAMETER>
```

The above syntax defines a new XML tag of type CONSTRAINT with a required ID attribute of value constraint-id, the parameter values for the constraint are defined by a PAREMETER tag with a required VALUE attribute value parameter-value. PARAMETER VALUE is different for different kinds of constraints.

Part 2: Defines the Relationships Among the Elements for the Host Domain

- **Role Hierarchy (Role-Role Relationship)**

Role hierarchy is represented as a set of INHERITES elements, each of which associates a role with its direct child role. For instance, a role hierarchy is represented by:

```
<! --Role hierarchy definition-- >
    <INHERITES FROM = ri  To  rj></INHERITES>
```

The above syntax defines a new XML tag of type INHERITES with a required FROM (role ri) and TO (role rj) attribute values which indicate role ri is a direct parent role of role rj.

- **Privilege Assignment (Privilege-Role Relationship)**

A privilege assignment assigns privilege to a role and it is represented by:

```
<! -- Privilege assignment definition-- >
        <PRIV-ASSIGN  ROLE = ti  PRIVILEGE=mi.></PRIV-ASSIGN>
```

The above syntax defines a new XML tag of type PRIV-ASSIGN with ROLE, and PRIVILEGE attributes in which role ti has a privilege mi.

- **Constraint Assignment (Constraint-Role Relationship)**

A constraint assignment assigns a set of role constraints to a role and it is represented by:

```
<! – Role constraint assignment definition-- >
    <CONS-ASSIGN ROLE = r1  CONSTRAINTS = c1,…….cm.  > </CONS-ASSIGN>
```

The above syntax defines a new XML tag of type CONS-ASSIGN with a required ROLE and CONSTRAINTS attributes in which role r1 has constraints of c1,…… cm .

- **Extended Parent Role Assignment (Role - its Extended Parent Role)**

An extended parent role assignment assigns a roles to be the extended parent role of a role in another local domain, and it is represented as:

```
<! – Extended parent role assignment definition-- >
        <EXTENDED PARENT ROLE epp= rj, rk > </EXTENDED PARENT ROLE >
```

The above syntax defines a new XML tag of type EXTENDED PARENT ROLE with a required epp attributes in which role rj is the extended parent role of role rk.

- **Extended Child Role Assignment (Role – its Extended Child Role)**

An extended child role assignment assigns a roles to be the extended child role of a role in another local domain, and it is represented as:

```
<! – Extended child role assignment definition-- >
        <EXTENDED CHILD ROLE ecp= rj, rk >  </EXTENDED CHILD ROLE >
```

The above syntax defines a new XML tag of type EXTENDED CHILD ROLE with a required ecp attributes in which role rj is the extended child role of role rk.

APPENDIX B

An Example of PRAC Security Policy for Enterprise Host Domain

This Appendix shows an example of PRAC security policy for an enterprise host domain according to the syntax defined in Appendix A.

```
< xml version= “1.0” >

<PRAC-MODEL TYPE= “PRAC1_POLICY”>

    < -- Local domain 1-- >

        <! -- Basic Elements -- >

            <! --Privilege set definition-- >
                <PRIVILEGE ID= “p1” > </PRIVILEGE>
                <PRIVILEGE ID= “p2” > </PRIVILEGE>
                <PRIVILEGE ID= “p3” > </PRIVILEGE>
                <PRIVILEGE ID= “p4” > </PRIVILEGE>
                <PRIVILEGE ID= “p5” > </PRIVILEGE>
            </ --Privilege set definition-- >


            <! –Role definition-- >
                <ROLE ID= “ A ”  > </ROLE>
                <ROLE ID= “ B ” >  </ROLE>
                <ROLE ID= “ C ”  > </ROLE>
            <ROLE ID= “ D ” >  </ROLE>
            <ROLE ID= “ E ”  > </ROLE>
            <ROLE ID= “ F ” >  </ROLE>
            <ROLE ID= “ G ”  > </ROLE>
        </ –Role definition-- >

        <! --Constraint set definition-- >
            <CONSTRAINT ID= “ C1 ” >
                    <PARAMETER VALUE= “ h1 h2”>< / PARAMETER></CONSTRAINT>
            <CONSTRAINT ID= “ C2” >
                    <PARAMETER VALUE= “ h3, h4”>< / PARAMETER></CONSTRAINT>
            <CONSTRAINT ID= “ C3” >
                    <PARAMETER VALUE= “ h5”>< / PARAMETER></CONSTRAINT>
            <CONSTRAINT ID= “ C4” >
                    <PARAMETER VALUE= “ h6, h7”>< / PARAMETER></CONSTRAINT>
            <CONSTRAINT ID= “ C5” >
                    <PARAMETER VALUE= “ h8, h9” >< / PARAMETER></CONSTRAINT>
            <CONSTRAINT ID= “ C6” >
                    <PARAMETER VALUE= “ h10”>< / PARAMETER></CONSTRAINT>
            <CONSTRAINT ID= “ C7” >
                    <PARAMETER VALUE= “ h11”>< / PARAMETER></CONSTRAINT>
            <CONSTRAINT ID= “ C8” >
                    <PARAMETER VALUE= “ h12”>< / PARAMETER></CONSTRAINT>
        </ --Constraint set definition-- >

</ -- Basic Elements -- >



< !-- Relationships of Elements -- >

    <! --Role hierarchy definition-- >
            <INHERITS FROM = “A”  To  “B” > </INHERITS>
            <INHERITS FROM = “A”  To  “C”></INHERITS>
            <INHERITS FROM = “B”  To  “E” > </INHERITS>
            <INHERITS FROM = “B”  To  “F”></INHERITS>
            <INHERITS FROM = “C”  To  “F” > </INHERITS>
        <INHERITS FROM = “D”  To  “F” > </INHERITS>
        <INHERITS FROM = “D”  To  “G”></INHERITS>
    </ --Role hierarchy definition-- >


    <! -- Privilege assignment definition-- >
        <PRIV-ASSIGN  ROLE= “A” PRIVILEGE=  “p1”></PRIV-ASSIGN>
        <PRIV-ASSIGN  ROLE= “B” PRIVILEGE=  “p2, p3”></PRIV-ASSIGN>
        <PRIV-ASSIGN  ROLE= “C” PRIVILEGE=  “p3”></PRIV-ASSIGN>
        <PRIV-ASSIGN  ROLE= “B” PRIVILEGE=  “p3”></PRIV-ASSIGN>
        <PRIV-ASSIGN  ROLE= “D” PRIVILEGE=  “p4, p5”></PRIV-ASSIGN>
    </ -- Privilege assignment definition-- >


    <! -- Constraint assignment definition-- >
        <CONS-ASSIGN ROLE= “A”  CONSTRAINTS= “C8”> </CONS-ASSIGN>
        <CONS-ASSIGN ROLE= “B”  CONSTRAINTS= “C7”> </CONS-ASSIGN>
        <CONS-ASSIGN ROLE= “C”  CONSTRAINTS= “C5”> </CONS-ASSIGN>
        <CONS-ASSIGN ROLE= “D”  CONSTRAINTS= “C4”> </CONS-ASSIGN>
        <CONS-ASSIGN ROLE= “E”  CONSTRAINTS= “C1, C3”> </CONS-ASSIGN>
        <CONS-ASSIGN ROLE= “F”  CONSTRAINTS= “C6”> </CONS-ASSIGN>
        <CONS-ASSIGN ROLE= “G”  CONSTRAINTS= “C2”> </CONS-ASSIGN>
    </ -- Constraint assignment definition-- >


    <! – Extended child role assignment definition-- >
       <EXTENDED CHILD ROLE ecp= D, I ></EXTENDED CHILD ROLE >
    </ – Extended child position role assignment definition-- >


  < /-- Relationships of Elements -- >
</-- Local domain 1-- >
```

```
<!-- Local domain 2-- >

  < !-- Basic Elements -- >

     <! --Privilege set definition-- >
          <PRIVILEGE ID= “p6” > </PRIVILEGE>
          <PRIVILEGE ID= “p7” > </PRIVILEGE>
     </ --Privilege set definition-- >


     <! –Role definition-- >
          <ROLE ID= “ I ”  > </ROLE>
          <ROLE ID= “ J ” >  </ROLE>
          <ROLE ID= “ K ”  > </ROLE>
     </ –Role definition-- >


     <! --Constraint set definition-- >
          <CONSTRAINT ID= “ Ca ”  >
               <PARAMETER VALUE= “ h14 ”>< / PARAMETER></CONSTRAINT>
          <CONSTRAINT ID= “ Cb ”  >
               <PARAMETER VALUE= “ h15”>< / PARAMETER></CONSTRAINT>
          <CONSTRAINT ID= “ Cc ”  >
               <PARAMETER VALUE= “ h16 h17”>< / PARAMETER></CONSTRAINT>
     </ --Constraint set definition-- >


  </ -- Basic Elements -- >


  <! -- Relationships of Elements -- >

     <! --Role hierarchy definition-- >
          <INHERITS FROM = “I”  To  “J” > </INHERITS>
          <INHERITS FROM = “I”  To  “K”></INHERITS>
     </ --Role hierarchy definition-- >
```

```
    <! -- Privilege assignment definition-- >
            <PRIV-ASSIGN  ROLE= “I” PRIVILEGE=  “p6, p7”></PRIV-ASSIGN>
            <PRIV-ASSIGN  ROLE= “J” PRIVILEGE=  “p7”></PRIV-ASSIGN>
    </ -- Privilege assignment definition-- >


    <! -- Constraint assignment definition-- >
            <CONS-ASSIGN ROLE= “I”  CONSTRAINTS= “Ca”> </CONS-ASSIGN>
            <CONS-ASSIGN ROLE= “J”  CONSTRAINTS= “Cc”> </CONS-ASSIGN>
            <CONS-ASSIGN ROLE= “K”  CONSTRAINTS= “Cb”> </CONS-ASSIGN>
    </ -- Constraint assignment definition-- >

    <! – Extended parent position role assignment definition-- >
            <EXTENDED PARENT ROLE epp= I, D ></EXTENDED PARENT ROLE >
     </ – Extended parent position role assignment definition-- >

  < /-- Relationships of Elements -- >

 </-- Local domain 2 -->

</PRAC-model>
```

APPENDIX C

Specifications of XML-based PRAC Security Policy for Foreign Domains

This Appendix defines the syntactic representation of the PRAC security policy for the foreign domain. The XML representation is comprised of two parts: Basic Elements and Relationships of Elements. The first part defines the elements of the PRAC model for the foreign domain including foreign-interfacing role, foreign-interfacing privilege, and credentials. The second part defines the relationships among the elements of the PRAC model for the foreign domain.

Part 1: Defines the Basic Elements of PRAC Model for the Foreign Domain

- **Define Foreign-Interfacing Role**

  The syntax of foreign-interfacing role defines an XML tag Foreign-Interfacing ROLE and an attribute ID which value is role-id.

```
<! – Foreign-Interfacing Role definition-- >
    <FOREIGN-Interfacing ROLE ID=role-id ></FOREIGN-Interfacing ROLE>
```

- **Define Foreign-Interfacing Privilege**

The syntax of foreign-interfacing privilege defines an XML tag Foreign-Interfacing PRIVILEGE and an attribute ID which value is privilege-id.

```
<! –Foreign-Interfacing Privilege set definition-- >
    <Foregin-Interfacing PRIVILEGE ID=privilege-id ></Foreign-Interfacing PRIVILEGE>
```

- **Define Credentials**

PRAC defines an XML-based credential format for each credential type. The syntax of credential defines an XML tag CREDENTIAL, an attribute ID which value is credential-id and an attribute TYPE which value is credential-type.

```
<! --Credential definition-- >
    <CREDENTIAL ID= credential-id, TYPE= credential-type >
    <SUBJECT-PPROPERTY ID=property-id (1)
    OPERATOR= subject-operator (1) Value=property-value(1)>< / SUBJECT-PROPERTY>
              ......
    < SUBJECT-PPROPERTY ID=property-id(n)
    OPERATOR= subject-operator(n) Value=property-value(n)>< / SUBJECT-PROPERTY>
  </CREDENTIAL>
```

Assume there are n subject properties and the subject properties in the credential type. The syntax of subject property defines an XML tag SUBJECT-PPROPERTY, an attribute ID which value is property-id, an OPERATOR attribute which value is subject-operator, such as ">" "<" "=", etc, and an attribute Value which is property-value.

Part 2: Defines the Relationships Among the Elements for the Foreign Domain

- **Foregin-Interfacing Role Hierarchy (Foreign-Interfacing Role-Role Relationship)**

  Foreign-interfacing role hierarchy is represented as a set of INHERITES elements, each of which associates a foreign-interfacing role with its direct child foreign-interfacing role.

  The syntax of foreign-interfacing role hierarchy defines a new XML tag of type INHERITES with a required FROM (foreign-interfacing role *ri*) and TO (foreign-interfacing role *rj*) attribute values which indicate foreign-interfacing role ri is a direct parent foreign-interfacing role of rj.

  ```
  <! –Foreign-Interfacing Role hierarchy definition-- >
          <INHERITES FROM = ri  To = rj></INHERITES>
  ```

- **Foreign-Interfacing Privilege Assignment (Foreign-Interfacing Role-Privilege Relationship)**

  Foreign-Interfacing-role privilege assignment assigns a set of foreign-interfacing privileges to a foreign-interfacing role. The syntax of foreign-interfacing privilege assignment defines an XML tag FOREIGN PRIV-ASSIGN with FOREIGN-Interfacing ROLE and PRIVILEGE attributes in which foreign role ti has privileges of pj,……pk.

  ```
  <! – Foreign-Interfacing Privilege assignment definition-- >
          <FOREIGN PRIV-ASSIGN FOREIGN-Interfacing ROLE = ti  PRIVILEGE= pj,……
          pk.></FOREIGN PRIV-ASSIGN>
  ```

- **Credential Assignment (Foreign-Interfacing Role-Credential Relationship)**

Credential assignment assigns a set of credential chains to a foreign-interfacing role. The syntax of credential assignment defines an XML tag of type CONS-ASSIGN with a required FOREIGN-Interfacing ROLE and CREDENTIALS attributes in which foreign role r1 has credential chains of c1 , cm ....and ck.

```
<! -- Credential assignment definition-- >
   <CONS-ASSIGN FOREIGN-Interfacing ROLE= r1
                              CREDENTIALS= c1 ∨ cm ∨......∨ ck></CONS-ASSIGN>
```

APPENDIX D

An Example of PRAC Security Policy for Foreign Domains

This Appendix shows an example of PRAC security policy for an foreign domain according to the syntax defined in Appendix C.

```
< xml version= “1.0” >

<PRAC-MODEL TYPE= “PRAC2_POLICY”>

    <! -- Basic Elements -- >

        <! –Foreign-Interfacing Privilege set definition-- >
            < FOREIGN-Interfacing PRIVILEGE ID= “p1” > </ FOREIGN-Interfacing PRIVILEGE>
            < FOREIGN-Interfacing PRIVILEGE ID= “p2” > </ FOREIGN-Interfacing PRIVILEGE>
            < FOREIGN-Interfacing PRIVILEGE ID= “p3” > </ FOREIGN-Interfacing PRIVILEGE>
            < FOREIGN-Interfacing PRIVILEGE ID= “p4” > </ FOREIGN-Interfacing PRIVILEGE>
            < FOREIGN-Interfacing PRIVILEGE ID= “p5” > </ FOREIGN-Interfacing PRIVILEGE>
            < FOREIGN-Interfacing PRIVILEGE ID= “p6” > </ FOREIGN-Interfacing PRIVILEGE>
        </ --Foreign-Interfacing Privilege set definition-- >


        <! –Foreign-Interfacing Role definition-- >
              < FOREIGN-Interfacing ROLE ID= “ G ”  > </FOREIGN-Interfacing ROLE>
              < FOREIGN-Interfacing ROLE ID= “ H ” >  </FOREIGN-Interfacing ROLE>
              < FOREIGN-Interfacing ROLE ID= “ I ”  >  </FOREIGN-Interfacing ROLE>
              < FOREIGN-Interfacing ROLE ID= “ J ” >   </FOREIGN-Interfacing ROLE>
              < FOREIGN-Interfacing ROLE ID= “ K ”  > </FOREIGN-Interfacing ROLE>
    </ – Foreign-Interfacing Role definition-- >


    <! --Credential set definition-- >
            <CREDENTIAL ID=“C1”  TYPE= “Power Enterprise” >
                  <SUBJECT-PPROPERTY ID= “Credit” OPERATOR= “=” Value= “5 ”>
                  </ SUBJECT PROPERTY>

                  <SUBJECT-PPROPERTY ID= “Balance” OPERATOR= “>=” Value= “200000”>
                  </ SUBJECT-PROPERTY>
            </CREDENTIAL>

            <CREDENTIAL ID=“C2”  TYPE= “Private sector”>
                  < SUBJECT-PPROPERTY ID= “Credit” OPERATOR= “>=”  Value= “4”>
                  </ SUBJECT-PROPERTY>
                  < SUBJECT-PPROPERTY ID= “Balance” OPERATOR= “>=” Value= “300000”>
                  </ SUBJECT-PROPERTY>
            </CREDENTIAL>
     </ --Credential set definition-- >
</ -- Basic Elements -- >



  < !-- Relationships of Elements -- >

    <! –Foreign-Interfacing role hierarchy definition-- >
            <INHERITS FROM = “G”  To  “H” ></INHERITS>
            <INHERITS FROM = “H”  To   “I”></INHERITS>
            <INHERITS FROM = “J”    To   “I” > </INHERITS>
            <INHERITS FROM = “K”   To  “H”></INHERITS>
    </ --Foreign-Interfacing role hierarchy definition-- >


    <! – Foreign-Interfacing Privilege assignment definition-- >
            <FOREIGN PRIV-ASSIGN  FOREIGN-Interfacing ROLE= “I”
                                         PRIVILEGE =  “p1  p2”></FOREIGN PRIV-ASSIGN>
            <FOREIGN PRIV-ASSIGN  FOREIGN-Interfacing ROLE= “I”
```

```
                                      PRIVILEGE = "p3  p4"></FOREIGN PRIV-ASSIGN>
                    <FOREIGN PRIV-ASSIGN  FOREIGN-Interfacing ROLE= "I"
                                      PRIVILEGE = "p5  p6"></FOREIGN PRIV-ASSIGN>
              </ -- Foreign-Interfacing Privilege assignment definition-- >


              <! -- Credential assignment definition-- >
                 <CONS-ASSIGN FOREIGN-ROLE= "G"  CREDENTIALS= "C1" > </CONS-ASSIGN>
                 <CONS-ASSIGN FOREIGN-ROLE= "H"  CREDENTIALS = "C2"> </CONS-ASSIGN>
                 <CONS-ASSIGN FOREIGN-ROLE= "I"  CREDENTIALS = "C1" > </CONS-ASSIGN>
                  <CONS-ASSIGN FOREIGN-ROLE= "J"  CREDENTIALS = "C2" > </CONS-ASSIGN>
                  <CONS-ASSIGN FOREIGN-ROLE= "K" CREDENTIALS = "C1" > </CONS-ASSIGN>
              </ -- Credential assignment definition-- >

           < /-- Relationships of Elements -- >

        </PRAC-model>
```

BIBLIOGRAPHY

[1] "Information Security Challenges in the Electric Power Industry," Symantec Enterprise Security, White Paper, Riptech, 2001, pp 1-10.

[2] A. Massoud, "Modeling and Control of Complex Interactive Networks," IEEE Control Systems Magazine, USA, 2002, pp 22-27.

[3] A. Massoud, "Towards Self-Healing Energy Infrastructure Systems," IEEE Computer Applications in Power, USA, 2001, pp 20-28.

[4] A. Massoud, "National Infrastructure as Complex Interactive Networks," Automation, Control, and Complexity: An Integrated Approach, Samad & Weyrauch (Eds.), John Wiley & Sons, 2000, pp 263-286.

[5] A. Menezes, P. Oorschot, and S. Vanstone, Handbook of Applied Cryptography, CRC Press, 1997.

[6] Fundamentals of Network Security, Cisco Networking Academy Program, Cisco press, 2004, pp 315-325.

[7] S. Bakhtiari, R. Safavi-Naini, and J. Pieprzyk, "Cryptographic Hash Functions: a Survey," Technical Report 95-09, Department of Computer Science, University of Wollongong, 1995.

[8] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman, "Role-Based Access Control Models," IEEE Computer, Vol. 29, No. 2, 1996, pp 38-47.

[9] E. Barka and R. Sandhu, " A Role-Based Delegation Model and Some Extensions," Proceding of 23rd National Information Systems Security Conference, 2000, pp 101-114.

[10] D. Ferraiolo, R. Sandhu, E. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed NIST Standard for Role-Based Access Control," ACM Transactions on Information and System Security, Vol. 4, No. 3, 2000, pp 224-274.

[11] S. Osborn, R. Sandhu, and Q. Munawer. "Configuring Role-based Access Control to Enforce Mandatory and Discretionary Access Control Policies," ACM Transactions on Information and System Security, Vol. 3, No. 2, 2000, pp 85-106.

[12] A. Kern, A. Schaad, and J. Moffett, "An administration Concept for the Enterprise Role-based Access Control Model," In Proceedings of the Eighth ACM Symposium on Access Control Models and Technologies (SACMAT'03), ACM Press, 2003, pp 3-11.

[13] R. Sandhu and V. Bhamidipati, "The ARBAC97 Model for Role-based Administration of Roles: Preliminary Description and Outline," Second ACM workshop on Role-Based-Access-Control, Fairfax, Virginia, USA, 1997, pp 41-54.

[14] M. Nyanchama and S. Osborn, "The Role Graph Mode," First ACM Workshop on Role-Based Access Control, Gaithersburg Maryland, 1995, pp 25-31.

[15] M. Nyanchama and S. Osborn, "Access Right Administration in Role-based Security Systems," In J. Biskup, M. Morgenstern, and C. E. Landwehr, editors, Database Security, VIII, Status and Prospects, Proceedings of the IFIP WG11.3 Working Conference on Database Security, North-Holland, 1994, pp 37-56.

[16] S. G. Akl and P.D. Taylor, “Cryptographic Solution to a Multilevel Security Problem,” In D. Chaum, R.L.Rivest, and A.T. Sherman, editor, Advanced in Cryptology, pp 237-249.

[17] S. G. Akl and P. D. Taylor, “Cryptographic Solution to a Problem of Access Control in a Hierarchy,” ACM Transaction on Computer Systems, Vol. 1, No. 3, 1983, pp 239-248.

[18] S. T. Mackinnon and P. D. Taylor, “An Optimal Algorithm for Assigning Cryptographic Keys to Control Access in a Hierarchy,” IEEE Transaction on Computer systems, Vol. C-34, No. 9, 1985, pp 797-802.

[19] S. T. MacKinnon and S. G. Akl, “New Key Generation Algorithms for Multilevel Security,” In Proceeding 1983 IEEE Symp on Security and Privacy, Oakland, CA, 1983, pp 72-78.

[20] T. C. Wu, T. S Wu, and W. H. He, “Dynamic Access Control Scheme Based on the Chinese Remainder Theorem,” Computer System Science and Engineering Vol. 2, 1995, pp 92-99.

[21] M. Verma, "XML Security: The XML Key Management Specification. XKMS Helps Make Security Manageable.", IBM Developer Works, Technology Report, 2004.

[22] F. Hirsch and M. Just, "XML Key Management (XKMS 2.0) Requirements," W3C Note, 2003, http://www.w3.org/TR/xkms2-req.

[23] T. Hardjono, B. Cain, and I. Monga, “Intra-domain Group Key Management Protocol,” Internet-Draft, 1998.

[24] J. H. Huang and S. M. Mykil, "A Highly Scalable Key Distribution Protocol for Large Group Multicast," In IEEE 2003 Global Communications Conference, 2003.

[25] C. K. Wong, M. Gouda, and S. S. Lam, "Secure Group Communications Using Key Graphs," In ACM SIGCOMM'98, ACM Press, 1998, pp 68-79.

[26] J. Crampton, "Specifying and Enforcing Constraints in Role-based Access Control," In Proceedings of the Eighth ACM Symposium on Access Control Models and Technologies (SACMAT 2003), Italy, 2003, pp 43-50.

[27] N. Li, Z. Bizri, and M. V. Tripunitara, "On Mutually-Exclusive Roles and Separation of Duty," Proceedings of the 11th ACM Conference on Computer and Communications Security, USA, 2004, pp 42-51.

[28] M. Blaze, J. Feignbaum, and J. Lacy, "Decentralized Trust Management," IEEE Symposium on Security and Privacy, Oakland, CA, MAY 1996, pp 17-28.

[29] M. Blaze, J. Feignbaum, and A. D. Keromytis, "KeyNote: Trust Management of Public-key Infrastructures," Security Protocals, 6th International Workshop, Cambridge UK, 1998, pp 59-63.

[30] M. Blaze, J. Feignbaum, J, Ioannidis, and A. Keromytis, "The KeyNote Trust Management System Version 2," Internet Draft RFC 2704, 1999.

[31] Extensible Markup Language (XML). http://www.w3.org/XML/.

[32] National Institute of Standards and Technology, Secure Hash Standard, FIPS 186-1, US Department of Commerce, April 1995.

[33] RFC 1810 - Report on MD5 Performance, http://www.faqs.org/rfcs/rfc1810.html

[34] N. A. Kofahi, T. Al-Somani, and K. Al-Zamil, "Performance Study of Some Symmetric Block Cipher Algorithms Under Linux Operating System," Journal of Discrete Mathematical Science & Cryptography, Vol. 7, No. 3, 2004, pp 359-370.