International Journal of Innovative Computing, Information and Control
Volume **8**, Number **9**, September **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 6219–6233

HIERARCHICAL INFORMATION-PROTECTED SYSTEM WITH MULTIPLE PREDECESSORS

Tsung-Chih Hsiao[1], Yu-Fang Chung[2], Tzer-Shyong Chen[3] and Gwoboa Horng[1]

[1]Department of Computer Science and Engineering
National Chung Hsing University
No. 250, Kuo Kuang Rd., Taichung 402, Taiwan
{ phd9408; gbhorng }@cs.nchu.edu.tw

[2]Department of Electrical Engineering
[3]Department of Information Management
Tunghai University
No. 181, Sec. 3, Taichung Port Rd., Taichung 40704, Taiwan
{ yfchung; arden }@thu.edu.tw

Received May 2011; revised September 2011

Abstract. *This work presents a hierarchical security model for controlling access requests in an information-protected system based on the Newton's interpolation polynomial. Users are partially sorted by priority, to form a hierarchical user-organization. The model is used not only to control the access requests but also to simplify and improve security efficiently. The application of polynomials to the key generation algorithm simplifies problems into linear joint equations, and so enhances performance. As such, several immediate predecessors are allowed to restore the unique polynomial for determining a shared immediate successor's key using individual key, respectively. That is, immediate predecessors can have common authority over the same immediate successors at minimum parameter storage cost.*

Keywords: Newton's interpolation polynomial, User hierarchy, Access control, Key generation algorithm

1. **Introduction.** As the Internet and its corresponding technologies have advanced rapidly, the sharing of resources over networks has become quite common. In practice, individual information and data in multilevel systems must be well protected; as such, management of resources and users through authorized access control devices is becoming increasingly important in information-protected systems. Resource access and the distribution of power are considered primary in organizations. For instance, confidential data such as official document system, order data, and decision-making system should be controlled, which are based on the limits of authority, that the hierarchy concept is inevitable. Indeed, having the distributed power to control the access to data is important. The proposed method utilizes hierarchy and Newton's interpolation polynomial for access control, has multi-nodes construct polynomials for the key, and applies personal key and parameter to data access so as to reduce the burden of the users. Different from the past methods, key generation algorithm simplifies problems into linear joint equations, and so enhances performance. By using parameters which are discarded after one-time computation to increase the difficulty in decrypting the key, the practical efficiency is promoted. For example, a large organization, such as a corporation or a university, is portioned out into many divisions or departments where each individual may be endowed with a number of duties, which are either disjointed or shared. For disjointed duties,

the duty department is well separated from others; in other words, the duty staff is allowed access only to the documents associated with the disjointed job. Likewise, for the shared ones, only the participating departments should be authorized to access secure documents, the access for which irrelevant departments should be disabled. Computer science on multi-user systems has become indispensable as Networks develop, especially on resource-sharing issues in computer communication systems. There has risen greater needs and responsibility for proper administration of documents under such a multi-user computer environment, where access control through authentication methods is needed to help filter authorized or qualified users to their corresponding applications. Such access controls lead to the formation of a user hierarchy, the problems of which first arose in multilevel organizations but has not been limited to military and governmental departments only, but also in private firms. Access controls in a user hierarchy have been used in database management systems, data communications and networks and numerous studies [1,6,9,10,13] on related applications have been published over the years.

An organization can be represented as a user hierarchy by using partially ordered sets. Users are divided into distinct security classes, C_1, $C_2, \cdots, C_n$, where n is the number of nodes in the user hierarchy. The security classes are then partially sorted using the binary relation "$\leq$" to classify the relationships among them. The example in Figure 1 below reveals such partially-ordered sets in a user hierarchy.

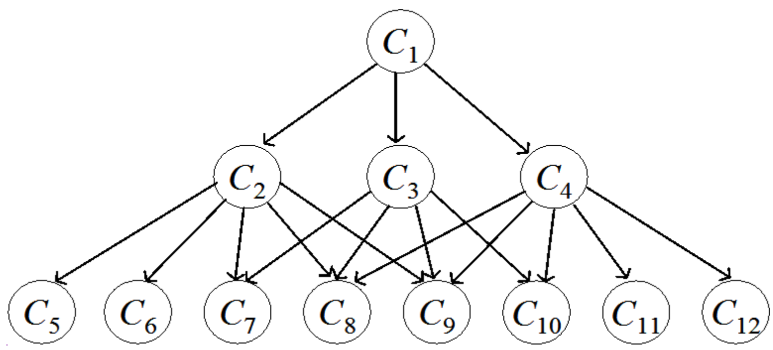

Figure 1. Partially-ordered sets in a multilevel information-protected system

Users are assigned different security-clearance levels based on the priority that is authorized through partial sorting. $C_j \leq C_i$ means that the priority of security class C_j is less than or equal to that of C_i. Also, C_i is said to be the predecessor of C_j, and C_j the successor of C_i. After being authorized, C_i can access the data of C_j but C_j cannot access those of C_i.

If no security class C_k exists such that $C_j \leq C_k \leq C_i$, then C_i is called the immediate predecessor of C_j and C_j the immediate successor of C_i. For convenience, the immediate predecessor and immediate successor are hereinafter abbreviated to IP and IS, respectively.

Recent studies on hierarchical key assignment schemes [2,3,11,14-16] have developed some important steps. In some of the studies, users in a hierarchical security model for controlling access requests in an information-protected system are assigned priorities such that the relationships among them can be linked and ranked. Such models have been able to provide an efficient means of controlling access requests and protecting users from having their data accessed illegally.

With Newton's interpolation polynomial to distribute the key to the controlled objects, the difference lies in generating keys with hierarchy for the computation of Newton's interpolation polynomial. Such a characteristic aims to enhance the security and to reinforce the possible factors in the hierarchy. As a lot of attacks utilize the loopholes among mathematical properties, disarranged relationships would be better. The advantages are listed as below.

(1) To promote efficacy and convenience. The keys are generated by multi-nodes constructing polynomials that the computations are easier and would not result in load for members, who simply use their own keys.
(2) To enhance security. When generating a key, nodes from different hierarchies are selected for Newton's interpolation polynomial so as to prevent it from attacks and to enhance the security.
(3) Easy to compute and restore the key. With Newton's interpolation polynomial, it is calculated with polynomials that several nodes could form a curve. It is therefore easy to restore the originally encrypted key.

The differences between the proposed method and the past methods appear on using Newton's interpolation polynomial to construct the encryption system. A polynomial is first selected, a node on the curve and the SK_i is randomly selected, and the hierarchy concept is utilized. For example, personal parameters and the parameter of a node which can be directly accessed are utilized to construct Newton's interpolation polynomial for key generation. In the stage of restoration, the original key is acquired by having multi-nodes to restore the decryption curve. The parameters which are discarded after one-time computation could increase the difficulty in mathematics that the original curve is not easily obtained for the decrypting the key.

The rest of this work is organized as follows. Section 2 reviews related studies. Section 3 elucidates the hierarchical security model. Section 4 evaluates security. Finally, Section 5 draws conclusions.

2. Review of Investigation on Hierarchical Systems.

2.1. **History of hierarchical systems.** With respect to bulk security classes, the key generation algorithm of the Akl-Taylar scheme [9] requires a large amount of memory to store the many keys of all users. The computing cost is high and the scheme is not very practical because of the large overheads [10]. Many other studies of access control management in hierarchical systems have been published, for example, the scheme [1] by C. C. Chang, R. J. Hwang, and T. C. Wu, which is based on the Newton's interpolation polynomial and a predefined one-way function that reveals the information required for key derivation within the parameter set by an individual. A comparison with the Akl-Taylar scheme shows that the space required to store the public parameters for the CHW scheme [1] is smaller; the key generation and derivation procedures are simpler, and the process is more efficient. However, two counterexamples [7] have been presented to demonstrate collisions in the CHW scheme. Two improved schemes [8] have also been determined to be insecure. Later in 2000, J. H. Wen, J. S. Sheu, and T. S. Chen proposed an improved scheme [5] that delivered better performance in a multilevel system without any collision when deriving keys.

2.2. **Incorrectness of the CHW scheme.** The preceding section briefly introduced the CHW scheme [1,12] and the controversy concerning performance and security leaks. The principle of the CHW scheme is to utilize a central authority (hereafter called CA for brevity) to generate and distribute security classes C_is' secret keys SK_is and public-parameter pairs $(P1_i, P2_i)$ in a user hierarchy. Before generating the secret keys, the CA sets the status of all security classes to "unmarked"; then, it generates the secret keys for all security classes using preorder tree traversal, and discloses the large prime P and the predefined one-way hash function $f(x)$ to all security classes.

Assume that a security class C_i has n ISs; then, $\Phi_i = \{C_{i,k}, k = 1, \ldots, n\}$ represents the set of all ISs subject to C_i. In Φ_i, $C_{i,k}$ is the kth IS of C_i, provided with a secret key $SK_{i,k}$ and a public-parameter pair $(P1_{i,k}, P2_{i,k})$. The key generation procedure endows each security class with an exclusive interpolation polynomial to control access requests between an IP and the IS. Through the Newton's interpolation polynomial [4], the CA designates a secret polynomial $H_i(x)$ using C_i's secret key $(0, SK_i)$ and all corresponding ISs' public-parameter pairs $(P1_{i,k}, P2_{i,k})$, where $k = 1, \ldots, n$. The polynomial therefore comprises ISs' keys. After $H_i(x)$ has been properly determined, the CA generates the secret key $SK_{i,k}$ of $C_{i,k}$, by applying the coefficient $a_{i,k}$ of the item x^k in $H_i(x)$ to $f(x)$, as follows:

$$SK_{i,k} = f(a_{i,k}) \bmod P$$

Similarly, C_i enables all ISs' secret keys to be determined from the exclusive secret polynomial. These n corresponding ISs' public-parameter pairs are used to restore the polynomial $H_i(x)$ produced during the key generation procedure, and then the coefficients of $H_i(x)$ are used in the predefined one-way function $f(x)$ to obtain the secret keys of all ISs.

The aforementioned procedure is the only means of restoring $H_i(x)$, so it is the only approach to determining the ISs' keys simultaneously. Although the CHW scheme performs efficiently, it involves collisions in a complex hierarchy. For instance, consider two entities of security classes C_i and C_l; both have the same security clearance and share the same n ISs. In deriving the shared ISs' keys, C_i firstly restores $H_i(x)$ using individual secret key $(0, SK_i)$ and the public-parameter pairs $(P1_{i,k}, P2_{i,k})$ of these n shared ISs, as follows:

$$H_i(x) = SK_i + a_{i,1}x + a_{i,2}x^2 + \cdots + a_{i,n}x^n \bmod P$$
$$SK_{i,k} = f(a_{i,k}) \bmod P, \text{ for } k = 1, \ldots, n$$

The same procedure for C_l to restore $H_l(x)$ using $(0, SK_l)$ and $(P1_{l,k}, P2_{l,k})$ of these same ISs is executed, as follows:

$$H_l(x) = SK_l + b_{l,1}x + b_{l,2}x^2 + \cdots + b_{l,n}x^n \bmod P$$
$$SK_{l,k} = f(b_{l,k}) \bmod P, \text{ for } k = 1, \ldots, n$$

In deriving secret key of the same IS, C_i substitutes $a_{i,k}$ in $H_i(x)$ and C_l substitutes $b_{l,k}$ in $H_l(x)$ into the predefined one-way function to, as follows:

$$f(a_{i,k}) = f(b_{l,k}) \bmod P, \text{ for } k = 1, \ldots, n$$

In fact, the secret keys of C_i and C_l differ; restated, the coefficients of the individual interpolation polynomials almost differ. Different input values do not yield the same key values via $f(x)$, and this is the collision in the CHW scheme. Figure 2 displays such a collision.

This means of constructing the interpolation polynomial might cause a serious security leak – collaboration from the ISs. That is, if a security class's ISs were to unite and attack their predecessor, the respective security class would be damaged [8], endangering the structure and security of the CHW scheme.

Based on a comprehensive survey of the related works, this paper proposes a security model against external and internal attacks that can also overcome the aforementioned collision and security leaks through a simple and efficient solution.

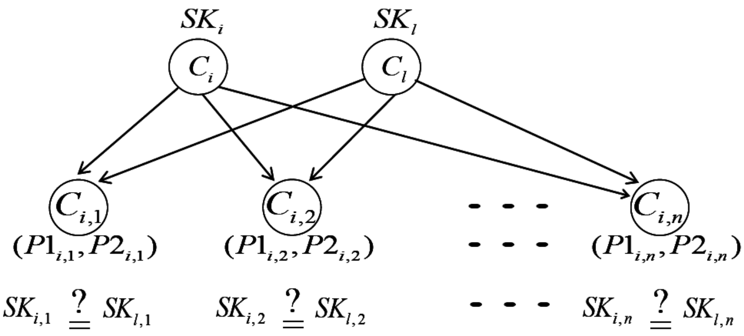

FIGURE 2. Events of collisions in the CHW scheme

3. Hierarchical Security Model in an Information-Protected System.

3.1. **Basic concept.** In a complex hierarchical environment with multiple relationships, let all security classes form a set S, in which the number of nodes is $|S|$. A set of terms IP, in which all members commonly share the same ISs, is defined as a similar-IP set. Assume that Q_L similar-IP sets are on the Lth security-clearance level. Denote these Q_L similar-IP sets as $\Psi_L = \{\Psi_{L,1}, \Psi_{L,2}, \ldots, \Psi_{L,Q_L}\}$ and denote the corresponding shared-IS sets subject to Ψ_L as $\varphi_L = \{\varphi_{L,1}, \varphi_{L,2}, \ldots, \varphi_{L,Q_L}\}$. The relationship between $\Psi_{L,j}$ and $\varphi_{L,j}$ is that the IPs in the former commonly have authority over the ISs in the latter. The secret keys of IPs in $\Psi_{L,j}$ are denoted $SK_{\Psi_{L,j},h}$, where $h = 1,\ 2, \ldots, |\Psi_{L,j}|$; those of ISs in $\varphi_{L,j}$ are denoted $SK_{\varphi_{L,j},k}$, where $k = 1, 2, \ldots, |\varphi_{L,j}|$.

Apart from the relationship between the similar-IP and shared-IS sets, another kind of relationship may exist between exclusive-IP and exclusive-IS sets. That is, if a security class C_i is the unique IP for some ISs, then such an IP is catalogued into the exclusive-IP set and the corresponding exclusive ISs are classified into the exclusive-IS set, Λ_i. A security class C_i can at the same time belong to an exclusive-IP set and/or to a similar-IP set. Table 1 defines the classified security classes, and Table 2 defines the given notations. Also, security classes are ranked by authorized priority, based on tracking by partially-ordered sorting. For instance, Table 3 classifies various partially-ordered sets, based on the user hierarchy in Figure 1.

3.2. **Procedure of generating and assigning keys.** In the scheme, each security class C_i is ranked according to priority to the security-clearance level and the key-generation algorithm is executed level by level recursively. The CA implements either exclusive-IP

TABLE 1. Definition of classified security classes

Item	Definition
C_i	Predecessor
C_j	Successor
IP	Immediate predecessor
IS	Immediate successor
non-IS	A successor C_j to the predecessor C_i, both interact via C_k such that $C_j \leq C_k \leq C_i$
Exclusive IP	The exclusive IP for an IS
Exclusive IS	An IS who is exclusively subject to the IP
Similar IP	An IP having authority over the same IS with other IPs
Shared IS	An IS who is subject to several IPs

Table 2. Definition of related notations

Notation	Definition
CA	Central Authority
S	The set of all users in the user hierarchy
L	Security-clearance level
$f(x)$	The predefined one-way function of the degree of $d+2$, where d is the maximal number of the ISs to an IP in the whole system [9]
P	A large prime number
$H_i(x)$	The interpolation polynomial of C_i
SK_i	The secret key of C_i
S_i	The secret parameter of C_i
$(R1_i, R2_i)$	The random parameter for CA in generating $H_i(x)$
$(P1_i, P2_i)$	The public-parameter pair of C_i
Ψ_L	Similar-IP set on the Lth security-clearance level
φ_L	Shared-IS set subject to Ψ_L
Φ_i	A set that collects all ISs subject to C_i, in which includes both kinds of exclusive and shared ISs
Λ_i	Exclusive-IS set subject to C_i

Table 3. Classification of various partially ordered sets shown as Figure 1

Similar-IP sets	$\Psi_{2,1} = \{C_2, C_3\}$ $\Psi_{2,2} = \{C_2, C_3, C_4\}$ $\Psi_{2,3} = \{C_3,\ C_4\}$
Shared-IS sets φ_2	$\varphi_{2,1} = \{C_7, C_8, C_9\}$ subject to $\Psi_{2,1}$ $\varphi_{2,2} = \{C_8, C_9\}$ subject to $\Psi_{2,2}$ $\varphi_{2,3} = \{C_8, C_9, C_{10}\}$ subject to $\Psi_{2,3}$
IS sets to exclusive IPs	$\Phi_2 = \{C_5, C_6, C_7, C_8, C_9\}$ subject to C_2 $\Phi_3 = \{C_7, C_8, C_9, C_{10}\}$ subject to C_3 $\Phi_4 = \{C_8, C_9, C_{10}, C_{11}, C_{12}\}$ subject to C_4
Exclusive-IS sets	$\Lambda_2 = \{C_5, C_6\}$ subject to C_2 $\Lambda_3 = \phi$ subject to C_3 $\Lambda_4 = \{C_{11}, C_{12}\}$ subject to C_4

or similar-IP sub-algorithms to generate the secret keys according to the class status that is classified to exclusive-IP or similar-IP sets. All IPs in a similar-IP set corresponding to the same shared-IS set share a random parameter, so that the different predecessors can restore a single interpolation polynomial using the individual parameter to derive the ISs' keys. Having made this property applied to our proposed scheme, the collusion shown in Figure 2 can now be solved.

3.2.1. *Key generation algorithm.* The CA generates and distributes the keys for each security class, as follows. After the initial settings as shown in Steps 1 and 2 are confirmed, Steps 3 and 4 of the key generation algorithm are to process the exclusive IPs, using the exclusive-IP sub-algorithm as described in Section 3.2.1.1. Steps 5 and 6 process the similar IPs, using the similar-IP sub-algorithm as described in Section 3.2.1.2.

Step 1a: Set the status of all nodes in the user hierarchy to "unmarked";

Step 1b: Denote the index of the security-clearance level as L; initially, set L to one for the highest security clearance.

Step 2a: Select an unmarked node from the security classes at the Lth security-clearance level;

Step 2b: Mark the selected node as C_i.

Step 3a: Determine the exclusive-IS set subject to C_i, and set it to Λ_i;

Step 3b: Execute the key generation and key assignment procedures using the exclusive-IP sub-algorithm.

Step 4: Repeat Steps 2a-3b until all nodes at the Lth security-clearance level have been marked.

Step 5a: Find the shared-IS sets that correspond to all similar-IP sets at the Lth security-clearance level, and designate the former sets as $\varphi_L = \{\varphi_{L,1}, \varphi_{L,2}, \ldots, \varphi_{L,Q_L}\}$ and the latter sets as $\Psi_L = \{\Psi_{L,1}, \Psi_{L,2}, \ldots, \Psi_{L,Q_L}\}$;

Step 5b: Designate the index of all similar-IP sets as j, and set the initial value of j to one.

Step 6a: Execute the key generation and key assignment procedures for the jth similar-IP set in Ψ_L using the similar-IP sub-algorithm to process $\Psi_{L,j}$;

Step 6b: Let $j = j + 1$. If $j \leq Q_L$, then return to Step 6a to process all similar-IP sets.

Step 7: When all nodes on the Lth security-clearance level have been marked; let $L = L + 1$. Then return to Step 2a to execute the key generation and assignment procedures for the next security-clearance level.

3.2.1.1. Exclusive-IP sub-algorithm.

The CA generates and distributes the secret keys $SK_{i,k}$ of the ISs $C_{i,k}$ and the secret parameter S_i of the corresponding exclusive IP C_i, as follows.

Step 1a: Generate a random pair of parameters $(R1_i, R2_i)$;

Step 1b: Generate the public-parameter pairs $(P1_{i,1}, P2_{i,1})$, $(P1_{i,2}, P2_{i,2}), \cdots, (P1_{i,|\Lambda_i|}, P2_{i,|\Lambda_i|})$ at random, and associate them with the exclusive ISs, $C_{i,1}, C_{i,2}, \ldots, C_{i,|\Lambda_i|}$ subject to C_i.

Step 2: Generate the following interpolation polynomial $H_i(x)$ based on the Newton's interpolation polynomial, using the parameters $(R1_i, R2_i)$, $(P1_{i,1}, P2_{i,1})$, $(P1_{i,2}, P2_{i,2}), \ldots, (P1_{i,|\Lambda_i|}, P2_{i,|\Lambda_i|})$.

$$H_i(x) = a_{i,0} + a_{i,1}x + a_{i,2}x^2 + \cdots + a_{i,|\Lambda_i|}x^{|\Lambda_i|} \pmod{P}$$

Step 3a: Generate the following secret keys $SK_{i,k}$ of $C_{i,1},\ C_{i,2}, \ldots, C_{i,|\Lambda_i|}$ by substituting the coefficients $a_{i,k}$ of x^k in $H_i(x)$ into the predefined one-way function $f(x)$.

$$SK_{i,k} = f(a_{i,k}) \pmod{P},\ \text{for } k = 1, \ldots, |\Lambda_i|$$

Step 3b: Generate the following secret parameter S_i of the IP C_i by substituting the secret key SK_i of C_i into $H_i(x)$.

$$S_i = H_i(SK_i)$$

Step 4a: Assign the secret keys $SK_{i,k}$ to the exclusive ISs subject to C_i for secret storage via a secure channel;

Step 4b: Assign the secret parameter S_i to C_i for secret storage via a secure channel.

Step 5: Declare $(P1_{i,1}, P2_{i,1})$, $(P1_{i,2}, P2_{i,2})$, $\cdots$, $(P1_{i,|\Lambda_i|}, P2_{i,|\Lambda_i|})$ publicly and destroy $(R1_i, R2_i)$ for security reasons.

3.2.1.2. Similar-IP sub-algorithm.

The CA generates and distributes the secret keys $SK_{\varphi_{L,j},k}$ of the shared ISs $C_{\varphi_{L,j},k}$ corresponding to $\Psi_{L,j}$ and the secret parameters $S_{\Psi_{L,j},v}$ of the similar IPs $C_{\Psi_{L,j},v}$ in $\Psi_{L,j}$, as follows.

Step 1: Select the representative of the security classes, $C_{\Psi_{L,j},1}$ from $\Psi_{L,j}$ corresponding to the shared-IS set $\varphi_{L,j}$, which includes $|\varphi_{L,j}|$ shared ISs.

Step 2a: Generate a pair of parameters $(R1_{\Psi_{L,j}}, R2_{\Psi_{L,j}})$ at random for the exclusive-IP set $\Psi_{L,j}$;

Step 2b: Generate the public-parameter pairs $(P1_{\varphi_{L,j},1}, P2_{\varphi_{L,j},1})$, $(P1_{\varphi_{L,j},2}, P2_{\varphi_{L,j},2})$, $\cdots$, $(P1_{\varphi_{L,j},|\varphi_{L,j}|}, P2_{\varphi_{L,j},|\varphi_{L,j}|})$ at random; and associate them with the shared ISs $C_{\varphi_{L,j},1}$, $C_{\varphi_{L,j},2}, \ldots, C_{\varphi_{L,j},|\varphi_{L,j}|}$ corresponding to $\Psi_{L,j}$.

Step 3: Generate the following interpolation polynomial $H_{L,j}(x)$ based on the Newton's interpolation polynomial, using $(R1_{\Psi_{L,j}},\ R2_{\Psi_{L,j}})$, $(P1_{\varphi_{L,j},1},\ P2_{\varphi_{L,j},1})$, $(P1_{\varphi_{L,j},2},$ $P2_{\varphi_{L,j},2})$, $\cdots$, $(P1_{\varphi_{L,j},|\varphi_{L,j}|},\ P2_{\varphi_{L,j},|\varphi_{L,j}|})$.

$$H_{L,j}(x) = a_{L,j,0} + a_{L,j,1}x + a_{L,j,2}x^2 + \cdots + a_{L,j,\left|\varphi_{L,j}\right|}x^{\left|\varphi_{L,j}\right|} \pmod P$$

Step 4a: Generate the following secret keys $SK_{\varphi_{L,j},k}$ of $C_{\varphi_{L,j},k}$ by substituting the coefficients $a_{L,j,k}$ of x^k in $H_{L,j}(x)$ into the predefined one-way function $f(x)$.

$$SK_{\varphi_{L,j},k} = f(a_{L,j,k}) \pmod P, \text{ for } k = 1, \ldots, |\varphi_{L,j}|$$

Step 4b: Generate the following secret parameters $S_{\Psi_{L,j},v}$ of the similar IPs $C_{\Psi_{L,j},v}$ in $\Psi_{L,j}$ by substituting the secret keys $SK_{\Psi_{L,j},v}$ of $C_{\Psi_{L,j},v}$ into $H_{L,j}(x)$.

$$S_{\Psi_{L,j},v} = H_{L,j}(SK_{\Psi_{L,j},v}), \text{ for } v = 1, \ldots, |\Psi_{L,j}|$$

Step 5a: Assign the secret keys $SK_{\varphi_{L,j},k}$ to all ISs $C_{\varphi_{L,j},k}$ in $\varphi_{L,j}$ for secret storage via a secure channel;

Step 5b: Assign the secret parameters $S_{\Psi_{L,j},v}$ to all IPs $C_{\Psi_{L,j},v}$ in $\Psi_{L,j}$ for secret storage via a secure channel.

Step 6: Declare $(P1_{\varphi_{L,j},1}, P2_{\varphi_{L,j},1})$, $(P1_{\varphi_{L,j},2}, P2_{\varphi_{L,j},2})$, $\cdots$, $(P1_{\varphi_{L,j},|\varphi_{L,j}|}, P2_{\varphi_{L,j},|\varphi_{L,j}|})$ publicly and destroy $(R1_{\Psi_{L,j}}, R2_{\Psi_{L,j}})$ for security reasons.

3.3. **Procedure of deriving keys.** Consider a case in which a security class C_i derives the secret key of the corresponding IS, $C_{i,k}$, using the individual secret key, SK_i. First, $C_{i,k}$ may be a member of an exclusive-IS set Λ_i or a shared-IS set $\varphi_{L,j}$, subject to the similar-IP set, $\Psi_{L,j}$ into which C_i is catalogued. The characteristics of an IS $C_{i,k}$, either exclusive or shared, determine for C_i the algorithm to be used after deriving the secret key of $C_{i,k}$. The key derivation procedure toward the IS for an IP C_i is as follows.

Step 1: Determine the relationship between the IP C_i and the corresponding IS $C_{i,k}$. If C_i is the exclusive IP of $C_{i,k}$, then execute Steps 2a-3; otherwise execute Steps 4a-5.

Step 2a: Determine the exclusive-IS set Λ_i subject to C_i;

Step 2b: Restore the following original interpolation polynomial $H_i(x)$ based on the Newton's interpolation polynomial, using the secret-parameter pair (SK_i, S_i) of C_i, and the public-parameter pairs $(P1_{i,1}, P2_{i,1})$, $(P1_{i,2}, P2_{i,2})$, $\cdots$, $(P1_{i,|\Lambda_i|}, P2_{i,|\Lambda_i|})$ of the exclusive ISs $C_{i,k}$.

$$H_i(x) = a_{i,0} + a_{i,1}x + a_{i,2}x^2 + \cdots + a_{i,|\Lambda_i|}x^{|\Lambda_i|} \pmod P$$

Step 3: Determine the following secret key $SK_{i,k}$ of the exclusive IS $C_{i,k}$ in Λ_i by substituting the coefficient $a_{i,k}$ of x^k in $H_i(x)$ into the predefined one-way function $f(x)$.

$$SK_{i,k} = f(a_{i,k}) \pmod P, \text{ for } k \in [1, |\Lambda_i|]$$

Step 4a: Determine the shared-IS set $\varphi_{L,j}$ corresponding to the similar-IP set $\Psi_{L,j}$ into which C_i is catalogued and denoted as $C_{\Psi_{L,j},v}$;

Step 4b: Restore the following original interpolation polynomial $H_{L,j}(x)$ based on the Newton's interpolation polynomial, using the secret-parameter pair ($SK_{\Psi_{L,j},v}$, $S_{\Psi_{L,j},v}$) of $C_{\Psi_{L,j},v}$, and the public-parameter pairs ($P1_{\varphi_{L,j},1}, P2_{\varphi_{L,j},1}$), ($P1_{\varphi_{L,j},2}$, $P2_{\varphi_{L,j},2}$), $\cdots$, ($P1_{\varphi_{L,j},|\varphi_{L,j}|}$, $P2_{\varphi_{L,j},|\varphi_{L,j}|}$) of the shared ISs $C_{\varphi_{L,j},k}$ in $\varphi_{L,j}$.

$$H_{L,j}(x) = a_{L,j,0} + a_{L,j,1}x + a_{L,j,2}x^2 + \cdots + a_{L,j,|\varphi_{L,j}|}x^{|\varphi_{L,j}|} \pmod P$$

Step 5: Determine the following secret key $SK_{\varphi_{L,j},k}$ of the shared IS $C_{\varphi_{L,j},k}$ in $\varphi_{L,j}$ by substituting the coefficient $a_{L,j,k}$ of x^k in $H_{L,j}(x)$ into the predefined one-way function $f(x)$.

$$SK_{\varphi_{L,j},k} = f(a_{L,j,k}) \pmod P, \quad \text{for } k \in [1, |\varphi_{L,j}|]$$

The C_i merely permits the corresponding ISs' secret keys to be derived; when accessing a non-IS, he must recursively execute the key derivation procedure, level by level, until the target node on the connected path is reached.

3.4. **Examples.** In this section, the example in Figure 3 show how the model involves key generation and key derivation procedures. The diagram, divided into three security-clearance levels, comprises nine security classes $C_1, C_2, \cdots, C_9$. Initially, let the prime number $P = 23$ and the predefined one-way function $f(x) = 5^x + 4x^3 + 7x^2 + 3x + 9 \pmod{23}$. The CA determines the system parameters, as presented in Table 4.

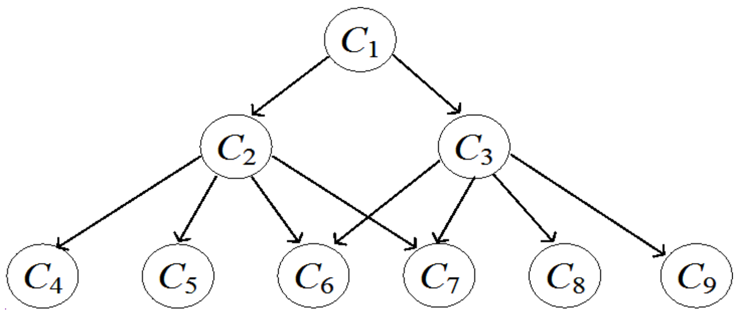

FIGURE 3. Illustration of a hierarchical organization

TABLE 4. System parameters

C_i	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
$(P1_i, P2_i)$	$(12,11)$	$(14,13)$	$(3,22)$	$(15,4)$	$(4,9)$	$(1,13)$	$(13,6)$	$(10,12)$	$(5,12)$
$(R1_i, R2_i)$	$(21,8)$	$(3,7)$	$(19,3)$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
(SK_i, S_i)	$(19,18)$	$(15,4)$	$(11,5)$	$(6,4)$	$(20,N)$	$(2,N)$	$(15,N)$	$(1,N)$	$(20,N)$
$(R1_{2,1}, R2_{2,1})$	$\cdots$	$(19,4)$	$(19,4)$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$S_{\Psi_{2,1},i}$	$\cdots$	18	12	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

Note: N indicates that no parameter is required.

3.4.1. *Example of the procedure of generating and assigning keys.*

3.4.1.1. Processing exclusive *IP*s for *CA*.

The procedures of key generation and assignment are executed level by level. Initially, for the first security-clearance level, the CA generates the parameters of exclusive IP C_1 so as to generate the keys of the corresponding exclusive ISs C_2 and C_3 in the exclusive-IS set $\Lambda_1 = \{C_2, C_3\}$ using $H_1(x)$, as follows.

Step 0: Randomly select a large prime P.

Step 1: Generate at random the following parameters.

$SK_1 = 19$, $(P1_1, P2_1) = (12, 11)$, and $(R1_1, R2_1) = (21, 8)$ to C_1, where $SK_1 \in Zp^*$.

$(P1_2, P2_2) = (14, 13)$ to C_2

$(P1_3, P2_3) = (3, 22)$ to C_3

Step 2: Generate the following interpolation polynomial $H_1(x)$, using the parameters $(R1_1, R2_1) = (21, 8)$, $(P1_2, P2_2) = (14, 13)$, and $(P1_3, P2_3) = (3, 22)$.

$$H_1(x) = 1 + 3x + 9x^2 \pmod{23}$$

Step 2.1: Generate the following SK_2 for C_2 by substituting the coefficient 3 of x in $H_1(x)$ into $f(x)$.

$$SK_2 = f(3) = 15$$

Step 2.2: Generate the following SK_3 for C_3 by substituting the coefficient 9 of x^2 in $H_1(x)$ into $f(x)$.

$$SK_3 = f(9) = 11$$

Step 2.3: Generate the following secret parameter S_1 for the exclusive IP C_1 by substituting $SK_1 = 19$ into $H_1(x)$.

$$S_1 = H_1(SK_1) = H_1(19) = 18$$

Step 3: Assign $SK_2 = 15$ to C_2, $SK_3 = 11$ to C_3, and $S_1 = 18$ to C_1 for secret storage so as to complete the assignment of secret keys and secret parameter.

Step 4: Declare $(P1_2, P2_2) = (14, 13)$ and $(P1_3, P2_3) = (3, 22)$ publicly and destroy $(R1_1, R2_1) = (21, 8)$ for security reasons.

Next, for the second security-clearance level, the CA generates the keys of the exclusive ISs C_4 and C_5 in the exclusive-IS set $\Lambda_2 = \{C_4, C_5\}$ corresponding to the exclusive IP C_2 using $H_2(x)$, as follows.

Step 1: Generate at random the following parameters.

$$(R1_2, R2_2) = (3, 7) \text{ to } C_2$$

$$(P1_4, P2_4) = (15, 4) \text{ to } C_4$$

$$(P1_5, P2_5) = (4, 9) \text{ to } C_5$$

Step 2: Generate the following interpolation polynomial $H_2(x)$, using the parameters $(R1_2, R2_2) = (3, 7)$, $(P1_4, P2_4) = (15, 4)$, and $(P1_5, P2_5) = (4, 9)$.

$$H_2(x) = 9 + 5x + 16x^2 \pmod{23}$$

Step 2.1: Generate the following SK_4 for C_4 by substituting the coefficient 5 of x in $H_2(x)$ into $f(x)$.

$$SK_4 = f(5) = 6$$

Step 2.2: Generate the following SK_5 for C_5 by substituting the coefficient 16 of x^2 in $H_2(x)$ into $f(x)$.

$$SK_5 = f(16) = 20$$

Step 2.3: Generate the following secret parameter S_2 for the IP C_2 by substituting $SK_2 = 15$ into $H_2(x)$.

$$S_2 = H_2(SK_2) = H_2(15) = 4$$

Step 3: Assign $SK_4 = 6$ to C_4, $SK_5 = 20$ to C_5, and $S_2 = 4$ to C_2 for secret storage so as to complete the assignment of secret keys and secret parameter.

Step 4: Declare $(P1_4, P2_4) = (15, 4)$ and $(P1_5, P2_5) = (4, 9)$ publicly and destroy $(R1_2, R2_2) = (3, 7)$ for security reasons.

With respect to the third security-clearance level, the CA generates the keys of the exclusive ISs C_8 and C_9 in the exclusive-IS set $\Lambda_3 = \{C_8, C_9\}$ corresponding to the exclusive IP C_3 using $H_3(x)$, as follows.

Step 1: Generate at random the following parameters.

$$(R1_3, R2_3) = (19, 3) \text{ to } C_3$$

$$(P1_8, P2_8) = (10, 12) \text{ to } C_8$$

$$(P1_9, P2_9) = (5, 12) \text{ to } C_9$$

Step 2: Generate the following interpolation polynomial $H_3(x)$, using the parameters $(R1_3, R2_3) = (19, 3)$, $(P1_8, P2_8) = (10, 12)$, and $(P1_9, P2_9) = (5, 12)$.

$$H_3(x) = 15 + 6x + 18x^2 \pmod{23}$$

Step 2.1: Generate the following SK_8 for C_8 by substituting the coefficient 6 of x in $H_3(x)$ into $f(x)$.

$$SK_8 = f(6) = 1$$

Step 2.2: Generate the following SK_9 for C_9 by substituting the coefficient 18 of x^2 in $H_2(x)$ into $f(x)$.

$$SK_9 = f(18) = 20$$

Step 2.3: Generate the following secret parameter S_3 for the exclusive IP C_3 by substituting $SK_3 = 11$ into $H_3(x)$.

$$S_3 = H_3(SK_3) = H_3(11) = 5$$

Step 3: Assign $SK_8 = 1$ to C_8, $SK_9 = 20$ to C_9, and $S_3 = 5$ to C_3 for secret storage so as to complete the assignment of secret keys and secret parameter.

Step 4: Declare $(P1_8, P2_8) = (10, 12)$ and $(P1_9, P2_9) = (5, 12)$ publicly and destroy $(R1_3, R2_3) = (19, 3)$ for security reasons.

3.4.1.2. Processing similar *IP*s for *CA*.

For the similar-IP set $\Psi_{2,1} = \{C_2, C_3\}$, the CA generates the secret keys SK_6 and SK_7 of the shared ISs C_6 and C_7 in the shared-IS set $\varphi_{2,1} = \{C_6, C_7\}$, and it generates the secret parameters $S_{\Psi_{2,1},1}$ and $S_{\Psi_{2,1},2}$ of the corresponding similar IPs C_2 and C_3 in $\Psi_{2,1}$ using $H_{2,1}(x)$, as follows.

Step 1: Generate at random the following parameters.

$$(R1_{2,1}, R2_{2,1}) = (19, 4) \text{ to } \Psi_{2,1}$$

$$(P1_6, P2_6) = (1, 13) \text{ to } C_6$$

$$(P1_7, P2_7) = (13, 6) \text{ to } C_7$$

Step 2: Generate the following interpolation polynomial $H_{2,1}(x)$, using the parameters $(R1_{2,1}, R2_{2,1}) = (19, 4)$, $(P1_6, P2_6) = (1, 13)$, and $(P1_7, P2_7) = (13, 6)$.

$$H_{2,1}(x) = 16 + 12x + 8x^2 \pmod{23}$$

Step 2.1: Generate the following SK_6 for C_6 by substituting the coefficient 12 of x in $H_{2,1}(x)$ into $f(x)$.

$$SK_6 = f(12) = 2$$

Step 2.2: Generate the following SK_7 for C_7 by substituting the coefficient 8 of x^2 in $H_{2,1}(x)$ into $f(x)$.

$$SK_7 = f(8) = 15$$

Step 2.3: Generate the following secret parameter $S_{\Psi_{2,1,1}}$ for C_2 in $\Psi_{2,1}$ by substituting $SK_2 = 15$ into $H_{2,1}(x)$.

$$S_{\Psi_{2,1,1}} = H_{2,1}(SK_2) = H_{2,1}(15) = 18$$

Step 2.4: Generate the following secret parameter $S_{\Psi_{2,1,2}}$ for C_3 in $\Psi_{2,1}$ by substituting $SK_3 = 11$ into $H_{2,1}(x)$.

$$S_{\Psi_{2,1,2}} = H_{2,1}(SK_3) = H_{2,1}(11) = 12$$

Step 3: Assign $SK_6 = 2$ to C_6, $SK_7 = 15$ to C_7, $S_{\Psi_{2,1,1}} = 18$ to C_2, and $S_{\Psi_{2,1,2}} = 12$ to C_3 for secret storage so as to complete the assignment of secret keys and secret parameters.

Step 4: Declare $(P1_6, P2_6) = (1, 13)$ and $(P1_7, P2_7) = (13, 6)$ publicly and destroy $(R1_{2,1}, R2_{2,1}) = (19, 4)$ for security reasons.

3.4.2. *Example of the key derivation procedure.*

3.4.2.1. Deriving keys for exclusive *IP*s.

Consider that the exclusive IP C_1 corresponds to the exclusive-IS set $\Lambda_1 = \{C_2, C_3\}$; C_1 executes the following procedure to derive the secret keys of the ISs C_2 and C_3 in Λ_1.

Step 1: Restore the following interpolation polynomial $H_1(x)$, using $(SK_1, S_1) = (SK_1, H_1(SK_1)) = (19, 18)$, $(P1_2, P2_2) = (14, 13)$, and $(P1_3, P2_3) = (3, 22)$.

$$H_1(x) = 1 + 3x + 9x^2 \pmod{23}$$

Step 2: Determine the following SK_2 by substituting the coefficient 3 of x in $H_1(x)$ into $f(x)$.

$$SK_2 = f(3) = 15$$

Step 3: Determine the following SK_3 by substituting the coefficient 9 of x^2 in $H_1(x)$ into $f(x)$.

$$SK_3 = f(9) = 11$$

Consider for example, the exclusive IP C_2 corresponding to the exclusive-IS set $\Lambda_2 = \{C_4, C_5\}$; C_2 executes the following procedure to derive the secret keys of the exclusive ISs, C_4 and C_5 in Λ_2.

Step 1: Restore the following interpolation polynomial $H_2(x)$, using $(SK_2, S_2) = (SK_2, H_2(SK_2)) = (15, 4)$, $(P1_4, P2_4) = (15, 4)$, and $(P1_5, P2_5) = (4, 9)$.

$$H_2(x) = 9 + 5x + 16x^2 \pmod{23}$$

Step 2: Determine the following SK_4 by substituting the coefficient 5 of x in $H_2(x)$ into $f(x)$.

$$SK_4 = f(5) = 6$$

Step 3: Determine the following SK_5 by substituting the coefficient 16 of x^2 in $H_2(x)$ into $f(x)$.

$$SK_5 = f(16) = 20$$

As in the procedures above, C_3 derives SK_8 and SK_9 similarly, by restoring $H_3(x)$ in the example of the exclusive IP C_3, corresponding to the exclusive-IS set $\Lambda_3 = \{C_8, C_9\}$.

3.4.2.2. Deriving keys for similar *IP*s.

Consider the similar-IP set $\Psi_{2,1} = \{C_2,\ C_3\}$, corresponding to the shared-IS set $\varphi_{2,1} = \{C_6, C_7\}$; both C_2 and C_3 in $\Psi_{2,1}$ execute the following procedures to derive the secret keys SK_6 and SK_7 of the shared ISs, C_6 and C_7 in $\varphi_{2,1}$.

Step 1: Restore the following interpolation polynomial $H_{2,1}(x)$, using either $(SK_2, S_{\Psi_{2,1},1}) = (SK_2, H_{2,1}(SK_2)) = (15, 18)$ of C_2 or $(SK_3, S_{\Psi_{2,1},2}) = (SK_3, H_{2,1}(SK_3)) = (11, 12)$ of C_3, $(P1_6, P2_6) = (1, 13)$, and $(P1_7, P2_7) = (13, 6)$.

$$H_{2,1}(x) = 16 + 12x + 8x^2 \pmod{23}$$

Step 2: Determine the following SK_6 by substituting the coefficient 12 of x in $H_{2,1}(x)$ into $f(x)$.

$$SK_6 = f(12) = 2$$

Step 3: Determine the following SK_7 by substituting the coefficient 8 of x^2 in $H_{2,1}(x)$ into $f(x)$.

$$SK_7 = f(8) = 15$$

4. **Evaluation of Security.**

In an actual medical network system, medical data, such as patient medical records, drug procurement, or medical official document system information, generally require confidential protection. They require confidential security and private access control. Since the applications are related to hierarchical access authority, the higher level would receive the larger power and more resources. The difference between the proposed method and the past methods appears on the overlap of access authority that various parent-nodes access to the same child-node. With Newton's interpolation polynomial, the proposed method utilizes the node parameter and the immediate child-node parameter for calculating $H_i(x)$ to be the access parameter of the overlapped child-node. It therefore could solve the problem of same parent-nodes. From Table 4, the keys 18 and 12 could solve the problem. Such a method could effectively prevent it from conspiracy and coordinated attack as well as promote the efficiency and security. Consider the possible means from attackers; the following cites the security strategy designed in the model to counter various attacks.

Attack 1: Suppose that a security class C_i has $|\Lambda_i|$ exclusive ISs, $C_{i,1}$, $C_{i,2}, \ldots$, and $C_{i,|\Lambda_i|}$, of which $C_{i,k}$ tries to reveal C_i's secret key SK_i. First, $C_{i,k}$ might test for recovering the polynomial $H_i(x)$ using the public-parameter pairs $(P1_{i,1}, P2_{i,1})$, $(P1_{i,2}, P2_{i,2}), \cdots$, $(P1_{i,|\Lambda_i|}, P2_{i,|\Lambda_i|})$, and then $C_{i,k}$ guesses SK_i through the equation $S_i = H_i(SK_i)$.

$H_i(x)$ is a $|\Lambda_i|$-degree polynomial; $C_{i,k}$ shall not be able to accurately recover $H_i(x)$ entirely by forcing the $|\Lambda_i|$ pairs of parameters, $(P1_{i,1}, P2_{i,1})$, $(P1_{i,2}, P2_{i,2})$, $\cdots$, $(P1_{i,|\Lambda_i|}$, $P2_{i,|\Lambda_i|})$. Also, if determining $H_i(x)$, $C_{i,k}$ shall not be able to obtain SK_i from the equation $S_i = H_i(SK_i)$, for which S_i is a secret parameter known only to C_i. Therefore, any exclusive ISs who schemes to reveal secret keys of IPs shall fail.

Attack 2: Consider the shared ISs, $C_{\varphi_{L,j},1}$, $C_{\varphi_{L,j},2}, \cdots, C_{\varphi_{L,j},|\varphi_{L,j}|}$ that are subject to the similar-IP set $\Psi_{L,j}$ for instance, of which $C_{\varphi_{L,j},k}$ intends to reveal the secret key $SK_{\Psi_{L,j},k}$ of the IP $C_{\Psi_{L,j},k}$ in $\Psi_{L,j}$. First, $C_{\varphi_{L,j},k}$ might test for recovering the polynomial $H_{L,j}(x)$ using the public-parameter pairs, $(P1_{\varphi_{L,j},1}, P2_{\varphi_{L,j},1})$, $(P1_{\varphi_{L,j},2}, P2_{\varphi_{L,j},2})$, $\cdots$, $(P1_{\varphi_{L,j},|\varphi_{L,j}|}$, $P2_{\varphi_{L,j},|\varphi_{L,j}|})$. Then, $C_{\varphi_{L,j},k}$ guesses $SK_{\Psi_{L,j},k}$ using the equation $S_{\Psi_{L,j},k} = H_{L,j}(SK_{\Psi_{L,j},k})$.

Since $H_{L,j}(x)$ is a $|\varphi_{L,j}|$-degree polynomial, it cannot be determined merely using the $|\varphi_{L,j}|$ public-parameter pairs, $(P1_{\varphi_{L,j},1}, P2_{\varphi_{L,j},1})$, $(P1_{\varphi_{L,j},2}, P2_{\varphi_{L,j},2})$, and $(P1_{\varphi_{L,j},|\varphi_{L,j}|}, P2_{\varphi_{L,j},|\varphi_{L,j}|})$. Even if recovering $H_{L,j}(x)$, $C_{\varphi_{L,j},k}$ cannot derive the secret key $SK_{\Psi_{L,j},k}$ from the equation $S_{\Psi_{L,j},k} = H_{L,j}(SK_{\Psi_{L,j},k})$, because $S_{\Psi_{L,j},k}$ is a secret parameter that only keeps the classes in the set $\Psi_{L,j}$ informed but keeps secret from all others, including $C_{\varphi_{L,j},k}$. Therefore, any IS in shared-IS set scheming to reveal IPs' secret key shall fail.

For Attack 1, the discussion is aimed at the case between the exclusive-IP set and the exclusive-IS set; as to Attack 2, the analysis is given on the case between the similar-IP set and the shared-IS set.

Attack 3: Security threats arise not only from an internal attacker but also from the external. As aforementioned, attackers firstly must recover the polynomial $H_i(x)$ or $H_{L,j}(x)$ and obtain the secret parameter S_i or $S_{\Psi_{L,j},k}$; only then can they determine the secret key. Because of information insufficiency on the part of the attackers, with regard to external attack, it is infeasible to force the secret key from public information.

Attack 4: Suppose that a security class C_i's ISs, exclusive or shared, conspire to determine the secret key SK_i or to recover the secret polynomial $H_i(x)$. Consider such a collusion from the exclusive ISs.

As the assumption in Attack 1, C_i has $|\Lambda_i|$ exclusive ISs, $C_{i,1}, C_{i,2}, \ldots, C_{i,|\Lambda_i|}$. Each of them is provided with a secret key $SK_{i,k}$, for $k = 1, 2, \cdots, |\Lambda_i|$.

Based on the Newton's interpolation polynomial, these $|\Lambda_i|$ IS s conspire to recover the following polynomial $H_i(x)$ using an unknown pair of parameter $(0, a'_{i,0})$ and the public-parameter pairs $(P1_{i,1}, P2_{i,1}), (P1_{i,2}, P2_{i,2}), \cdots, (P1_{i,|\Lambda_i|}, P2_{i,|\Lambda_i|})$.

$$H_i(x) = a'_{i,0} + A_1(a'_{i,0})x + A_2(a'_{i,0})x^2 + \cdots + A_{|\Lambda_i|}(a'_{i,0})x^{|\Lambda_i|} \pmod P$$

where $A_k(a'_{i,0})$, for $k = 1, 2, \cdots, |\Lambda_i|$, forms a linear polynomial in the case of without being informed of the variable $a'_{i,0}$, i.e., $A_1(a'_{i,0}) = b_1 a'_{i,0} + b_0$, where b_1 and b_0 are integers.

If $|\Lambda_i|$ is the maximal number of the immediate successors of security class in the whole system, then the degree of the one way function $f(x)$ shall be $|\Lambda_i| + 2$; these collusive participants may construct the following equation $f(A_k(a'_{i,0}))$ using the coefficients $A_k(a'_{i,0})$ in $H_i(x)$ and their secret keys $SK_{i,k}$, for $k = 1, 2, \cdots, |\Lambda_i|$.

$$\begin{aligned} f(A_k(a'_{i,0})) &= SK_{i,k} \\ &= n_{k,(|\Lambda_i|+2)} {a'}_{i,0}^{|\Lambda_i|+2} + n_{k,(|\Lambda_i|+1)} {a'}_{i,0}^{|\Lambda_i|+1} + \cdots + n_{k,1} a'_{i,0} + n_{k,0} \pmod P \end{aligned}$$

where $SK_{i,k}$ and $n_{k,(|\Lambda_i|+2)}, n_{k,(|\Lambda_i|+1)}, \cdots, n_{k,1}, n_{k,0}$ are all known integers.

The means of recovering $H_i(x)$ is executed by solving $a'_{i,0}$ from these $|\Lambda_i|$ equations. In constructing the $|\Lambda_i|$ equations, there are $|\Lambda_i| + 2$ unknown parameters; the obtainable information is insufficient to determining $a'_{i,0}$. Consequently, the collusion fails to restore $H_i(x)$ and faces even greater difficulty to obtain C_i's secret key SK_i.

5. Conclusions.

The developed model, based on the Newton's interpolation polynomial, not only achieves to control access requests but also simplifies and improves security efficiently. The application of polynomials in the key generation algorithm simplifies problems into linear joint equations, thus enhancing performance. Even the user hierarchy is re-organized; the CA only needs a downward search to update. The proposed model enables the security classes in a similar-IP set to have common authority over the same ISs using individual keys without requiring favors from either other security classes at the same security-clearance level or the predecessor. Additionally, no successor can determine the secret key of its predecessor through attacks or guesses.

Acknowledgment. This work was supported partially by National Science Council of Taiwan under Grants NSC 101-2410-H-129-001.

REFERENCES

[1] C. C. Chang, R. J. Hwang and T. C. Wu, Cryptographic key assignment scheme for access control in a hierarchy, *Information Systems*, vol.17, no.3, pp.243-247, 1992.
[2] C.-H. Lin, W. Lee and Y.-K. Ho, An efficient hierarchical key management scheme using symmetric encryptions, *The 19th International Conference on Advanced Information Networking and Applications*, vol.2, no.28-30, pp.399-402, 2005.
[3] C. Yang, C. Li and R. Cheung, Cryptographic key management solution in a role hierarchy, *Canadian Conference on Electrical and Computer Engineering*, vol.1, no.2-5, pp.575-578, 2004.
[4] D. E. Knuth, *The Art of Computer Programming, Vol.2: Seminumerical Algorithms*, 3rd Edition, Addison-Wesley, MA, 1997.
[5] J. H. Wen, J. S. Sheu and T. S. Chen, Cryptographic key assignment scheme for overcoming the incorrectness of the CHW scheme, *Proc. of IEE Communications*, vol.148, no.4, pp.260-264, 2001.
[6] J. K. Jan and Y. M. Tseng, Two integrated schemes of user authentication and access control in a distributed computer network, *Proc. of IEE Computers and Digital Techniques*, vol.145, no.6, pp.419-424, 1998.
[7] K. Tan, S. Gu and H. Zhu, Correctness of CHW cryptographic key assignment scheme in a hierarchy, *Proc. of IEE Computers and Digital Techniques*, vol.146, no.4, pp.217-218, 1999.
[8] M. S. Hwang, C. C. Chang and W. P. Yang, Modified Chang-Hwang-Wu access control scheme, *IEE Electronics Letters*, vol.29, no.24, pp.2095-2096, 1993.
[9] S. G. Akl and P. D. Taylor, Cryptographic solution to a problem of access control in a hierarchy, *ACM Transactions on Computer Systems*, vol.1, no.3, pp.239-248, 1983.
[10] S. J. MacKinnon, P. D. Taylor, H. Meijer and S. G. Akl, An optimal algorithm for assigning cryptographic keys to control access in a hierarchy, *IEEE Trans. on Computers*, vol.34, no.9, pp.797-802, 1985.
[11] S.-Y. Wang and C.-S. Laih, Merging: An efficient solution for a time-bound hierarchical key assignment scheme, *IEEE Trans. on Dependable and Secure Computing*, vol.3, no.1, pp.91-100, 2006.
[12] T. S. Chen, K. S. Huang and Y. F. Chung, Modified cryptographic key assignment scheme for overcoming the incorrectness of the CHW scheme, *Applied Mathematics and Computation*, vol.159, no.1, pp.147-155, 2004.
[13] W. P. Lu and M. K. Sundareshan, A model for multilevel security in computer networks, *IEEE Transactions on Software Engineering*, vol.16, no.6, pp.647-659, 1990.
[14] M. J. Atallah, M. Blanton, N. Fazio and K. B. Frikken, Dynamic and efficient key management for access hierarchies, *ACM Trans. on Information and System Security*, vol.12, no.3, 2009.
[15] Y. F. Chung, H. H. Lee, F. Lai and T. S. Chen, Access control in user hierarchy based on elliptic curve cryptosystem, *Information Sciences*, vol.178, no.1, pp.230-243, 2008.
[16] D. Giri and P. D. Srivastava, A cryptographic key assignment scheme for access control in poset ordered hierarchies with enhanced security, *International Journal of Network Security*, vol.7, no.2, pp.223-234, 2008.