A CRYPTOGRAPHIC KEY ASSIGNMENT SCHEME FOR IMPROVING THE INCORRECTNESS OF CHW SCHEME

[1;¤]Jyh-Horng Wen; [1]Jeng-Shin Sheu, [2]Yu-Fang Chung, and [2]Tzer-Shyong Chen
[1]Department of Electrical Engineering
National Chung Cheng University, Chia-Yi, Taiwan 621, ROC
E-mail: wen@ee.ccu.edu.tw
[2]Department of Computer Science and Information Engineering
Dayeh University, Taiwan, R.O.C.
[¤]Correspondence addressee

ABSTRACT

Based on the Newton interpolation method and a prede¯ned one-way function, a cryptographic key assignment scheme, called the CHW scheme, in a user hierarchy was presented by Chang et al. in 1992 [4]. The CHW scheme solved the need of the dramatic storage in Akl-Taylor scheme [1]. However, two counter examples presented in [6] prove the incorrectness of CHW scheme, and further two modi¯ed versions of CHW scheme [8] are proven to be insecure as well. Owing to the above-mentioned problems, in this paper a simple scheme is proposed to solve the incorrectness and to enhance the security of CHW scheme.

1. INTRODUCTION

In an information protection system, the security of access control is very important. There are many schemes [1-5], which have been proposed to discuss about the access control in a user hierarchy. A user hierarchy can be represented by a partially order set (poset). In such hierarchy, the users are divided into di®erent security classes named $C_1; C_2; :::; C_n$, where n is the number of nodes in the user hierarchy. Figure 1 shows an example of the poset in a user hierarchy. According to the partially order **5** , the relationship among the security classes is presented. For instance, C_j **5** C_i means that the users in C_i have the authority to access the data in C_j, but the opposite is not allowed. Under such a relationship, C_i is called a predecessor of C_j, and C_j a successor of C_i. Moreover, if there does not exist any other security class C_k such that C_j **5** C_k **5** C_i, then C_j is an immediate successor of C_i and C_i is an immediate predecessor of C_j. For simplicity, throughout this paper we use the abbreviations IS and IP to denote an immediate successor and an immediate predecessor, respectively.

For a large number of security classes, the key generation algorithm of Akl-Taylor scheme [1] has been proved infeasible [5]. In order to improve the disadvantage of the Akl-Taylor scheme, a cryptographic key assignment scheme [4], called the CHW scheme based on the Newton interpolation method and a prede¯ned one-way function, was presented. Compared with Akl-Taylor scheme, the storage required for the public parameters in CHW scheme is much smaller, and moreover the process in generating and deriving keys becomes simple and e± cient. However, two counter examples proposed recently in [6] show that the CHW scheme [4] is incorrect and its two modi¯ed versions [8] are insecure. In this paper, a scheme is presented not only to improve the incorrectness of CHW scheme, but also to enhance the ability of defending against attacks.

2. THE INCORRECTNESS AND WEAKNESS OF CHW SCHEME

In this section, a brief introduction to CHW scheme [4] is given, and its incorrectness and weakness are presented subsequently. For any security class C_i in a user hierarchy, both of his secret key SK_i and his public-parameter pair $(P1_i; P2_i)$ are generated and distributed by the central authority (CA). A large prime number P and a prede¯ned one-way function f are public to all security classes in the user hierarchy by CA. Throughout this paper, we suppose that a security class C_i has k_i ISs, denoted by $\copyright_i = \{C_{i;j}; j = 1; 2; :::; k_i\}$, for which $SK_{i;j}$ and $(P1_{i;j}; P2_{i;j})$ denote the secret key and the pair of public parameters for the jth IS $C_{i;j}$; $j = 1; 2; :::, k_i$, respectively. According to the concept of Newton interpolation method [9], CA can construct an interpolating polynomial for each security class C_i in a user hierarchy, denoted as $H_i(x)$ of degree k_i, over the Galois ¯eld GF(P) by interpolating the following $k_i + 1$ points: $(0; SK_i)$ and the k_i public-parameter pairs $(P1_{i;j}; P2_{i;j})$, $j = 1; 2; :::, k_i$. Then the secret key $SK_{i;j}$ for the jth IS $C_{i;j}$ of C_i is generated by

$$SK_{i;j} = f(a_j) \pmod{P}; \quad (1)$$

where a_j is the coe± cient of the term x^j in $H_i(x)$:

At the beginning of the key generation process, all security classes in the user hierarchy are unmarked, and then traversed by the preorder way. The key-generation procedure of CHW scheme is described in detail in the following.

Step 1:
Get an unmarked node C_i from the user hierarchy by preorder traversal.

Step 2:
If C_i is a leaf node, that is, $k_i = 0$, then mark C_i and return to Step 1.

Step 3:
Let $C_{i;1}; C_{i;2}; :::; C_{i;m_i}$ be unmarked ISs of C_i and $C_{i;m_i+1}; C_{i;m_i+2}; :::; C_{i;k_i}$ be marked ones.

Step 4:
If C_i is the root node, that is, C_i has no predecessor, then go to Step 5; else go to Step 6.

Step 5:
(5a) Randomly select an integer between 1 and $P ¡ 1$, denoted as SK_i. Then assign SK_i to be the secret key of C_i and mark C_i.
(5b) Randomly select a polynomial of degree k_i over GF(P), denoted as

$$H_i(x) = SK_i + a_1x + a_2x^2 + ::: + a_{k_i}x^{k_i} \pmod{P};$$

where $a_1; a_2; :::; a_{k_i}$ are k_i distinct integers between 1 and $P ¡ 1$.
(5c) Go to Step 7.

Step 6:
(6a) Randomly select m_i integer pairs $(P1_{i;j}; P2_{i;j})$, $j = 1; 2; :::, m_i$, between 1 and P, such that all $P1_{i;t}$ for $t = 1; 2; :::; k_i$ are distinct.
(6b) By the Newton's interpolating method, an interpolating polynomial $H_i(x)$ of degree k_i on the $k_i + 1$ points: $(0; SK_i); (P1_{i;1}; P2_{i;1}); (P1_{i;2}; P2_{i;2}); :::;$ and $(P1_{i;k_i}; P2_{i;k_i})$ over GF(P) can be constructed as

$$H_i(x) = SK_i + a_1x + a_2x^2 + ::: + a_{k_i}x^{k_i} \pmod{P}:$$

Step 7: Generate the secret keys $SK_{i;j}$ of C_i's ISs, which are still unmarked, according to equation (1), and then mark $C_{i;j}$ for $j = 1; 2; :::; m_i$:
Step 8: Repeat from Step 1 until all nodes of the user hierarchy are marked. ¥

In the key derivation procedure, a security class C_i can reconstruct the interpolating polynomial $H_i(x)$ by his secret key SK_i and the k_i pairs of public parameters of his ISs, and then use $H_i(x)$ and the prede¯ned one-way function to derive the secret keys of all his ISs. For any non-immediate successor, C_i can derive the secret key by performing the key-derivation procedure iteratively. Since no one can reconstruct $H_i(x)$ only by the public parameters of C_i's ISs, the secret key of any security class cannot be derived by conspiratorial.

In the sequel, we discuss the incorrectness of CHW scheme. Let the set of security classes $Â = \{C_i; C_{i+1}; :::; C_{i+d_i-1}\}$ have the same security clearance; that is, all the elements of the set are on the same level of a user hierarchy. Suppose that the ¯rst q ISs of each security class in Â are the same . Because the keys are generated by preorder traversal, the ¯rst security class C_i in Â determines the secret keys and public-parameter pairs for the ¯rst q ISs, shared by all security classes in Â. Then these q ISs are marked. That is, C_i uses the points $(0; SK_i)$ and the k_i public-parameter pairs of his ISs, to reconstruct the following interpolating polynomial, denoted as

$$H_i(x) = SK_i + a_{i;1}x + a_{i;2}x^2 + ::: + a_{i;k_i}x^{k_i} \pmod{P};$$

Then C_i uses the k_i coe± cients, $a_{i;1}; a_{i;2}; :::; a_{i;k_i}$, to compute the secret keys of his ISs according to (1). When it comes to the other security classes in Â, their interpolating polynomials are given by

$$H_j(x) = SK_j + a_{j;1}x + a_{j;2}x^2 + ::: + a_{j;k_j}x^{k_j} \pmod{P};$$

for $j = i + 1; i + 2; :::; i + d_i - 1$. The q coe± cients $a_{k;1}; a_{k;2}; :::; a_{k;q}$ of $H_k(x)$; for $k = i; i+1; :::; i+d_i-1$; are used to generate the secret keys of the q shared ISs. Accordingly if each security class in Â wants to generate identical secret keys for their q shared ISs, then for each $r = 1; 2; :::; q$ the following equations

$$f(a_{i;r}) = f(a_{j;r}) \pmod{P} \quad \text{for all } i \neq j$$

must be satis¯ed. However they are not held in general due to distinct secret keys of security classes in Â. This is the incorrectness of CHW scheme and leads the CHW scheme unusable. Moreover, in CHW scheme, the secret key of a certain security class is susceptible to being broken if all his ISs are collaborated. Therefore any IP may be broken if all his ISs are united to invade their predecessor. In the next section, a simple and e®ective scheme will be presented to solve the two problems.

3. OUR PROPOSED SCHEME TO IMPROVE CHW SCHEME

Assume that a central authority (CA) is responsible of generating and distributing the secret key SK_i

and public-parameter pair $(P1_i; P2_i)$ for each security class C_i in the user hierarchy. Let P be a large prime number and f be a predefined one-way function. Both P and f are made public to all security classes in the user hierarchy by CA. Moreover, to keep from collaborative attacks [8], any secret key SK will be substituted with its corresponding pretending secret key SK', generated from the predefined function f.

$$SK' = f(SK) \tag{2}$$

3.1 THE BASIC IDEA OF THE PROPOSED SCHEME

Firstly, we assume that for a set of security classes, all the security classes in this set have the same security clearance. A set of IPs is called a similar IP set if all security classes of which simultaneously share a number of ISs. Suppose that there are Q_L similar IP sets in the Lth security-clearance level. We use $^a_L = \{^a_{L;1}; ^a_{L;2}; \ldots; ^a_{L;Q_L}\}$ to denote the Q_L similar IP sets and assume that the jth similar IP set $^a_{L;j}$ of ^a_L contains $N_{^a_{L;j}}$ IPs for $j = 1; 2; \ldots; Q_L$. Every similar IP set corresponds to a set of ISs, which is called a shared IS set. For simplicity, we use $'_{L;j}$ to denote the shared IS set corresponding to the similar IP set $^a_{L;j}$. In addition, let $'_L = \{'_{L;1}; '_{L;2}; \ldots; '_{L;Q_L}\}$ denote the Q_L shared IS sets corresponding to the similar IP sets $^a_L = \{^a_{L;1}; ^a_{L;2}; \ldots; ^a_{L;Q_L}\}$, and the number of ISs in $'_{L;j}$ be $N_{'_{L;j}}$ for $j = 1; 2; \ldots; Q_L$. That is, the $N_{'_{L;j}}$ security classes in $'_{L;j}$ are shared by each IP in $^a_{L;j}$.

Let $¤_j$ be a set containing $N_{¤_j}$ security classes. A security class C_j is called the exclusive IP with respect to the set $¤_j$, if C_j is the only IP that exclusively shares the $N_{¤_j}$ security classes in $¤_j$. For simplicity, we call $¤_j$ the exclusive IS set with respect to the exclusive IP, C_j. Previously, we suppose that a security class C_j in the user hierarchy has k_j ISs, denoted by $©_j = \{C_{j;t}; t = 1; 2; \ldots; k_j\}$. It is observed that k_j equals $N_{¤_j}$, if C_j does not belong to any similar IP set. For a good comprehension of the above-defined terminologies, an illustration for the user hierarchy in Figure 1 is given. In Figure 1, $\{C_1\}$, $\{C_2; C_3; C_4\}$, and $\{C_5; C_6; \ldots; C_{18}\}$ belong to the first, second, and third security-clearance level, respectively. The illustration for the second security-clearance level is shown at Table 1. Apparently, a certain security class may belong to a similar IP set and an exclusive IP at the same time. For example, C_2 belongs to $^a_{2;1}$ the first similar IP set of the second security-clearance level, and is also the exclusive IP of the exclusive IS set $¤_2$.

In our proposed scheme, for each security-clearance level, the security classes of similar IP sets and exclusive IPs are done separately by different algorithms. Accordingly, while we construct interpolating polynomials, the ISs of any node C_i in the user hierarchy are classified into two parts, if exists. The first part is the shared IS set corresponding to the similar IP set to which C_i belongs, and the second part is the exclusive IS set whose exclusive IP is C_i. Consider a certain similar IP set $^a_{L;j}$ with respect to the shared IS set $'_{L;j}$. The criteria for the key-generation scheme is that each security class in $^a_{L;j}$ can only use his own secret key, without any secret key of the other peers in $^a_{L;j}$, on deriving secret keys of their shared ISs in $'_{L;j}$. And importantly, it must satisfy that any IP in $^a_{L;j}$ cannot use the secret keys of the shared IS set $'_{L;j}$ to derive any secret key of the other peers in $^a_{L;j}$. In the next section, we propose a simple and effective scheme satisfying the above two points. The proposed scheme is based on the combination of Lagrange polynomial [11] and Newton interpolation method [9]. In the sequel, $\{SK_{^a_{L;j};k}; k = 1; 2; \ldots; N_{^a_{L;j}}\}$ are used to denote the secret keys of the $N_{^a_{L;j}}$ IPs in $^a_{L;j}$. About the basic idea of the Lagrange polynomial, we would like to consider the product of factors first given by

$$£_{^a_{L;j}}(x) = \prod_{k=1}^{N_{^a_{L;j}}} (x - SK'_{^a_{L;j};k}); \tag{3}$$

which is related to the $N_{^a_{L;j}}$ pretending secret keys $\{SK'_{^a_{L;j};k}; k = 1; 2; \ldots; N_{^a_{L;j}}\}$. The function $£_{^a_{L;j}}(x)$ is a polynomial of $N_{^a_{L;j}}$ orders and becomes zero at $x = SK'_{^a_{L;j};1}; SK'_{^a_{L;j};2}; \ldots;$ and $SK'_{^a_{L;j};N_{^a_{L;j}}}$. If $£_{^a_{L;j}}(x)$ is divided by $(x - SK'_{^a_{L;j};i})$, the resulting function, defined to be

$$V_i(x) = \frac{£_{^a_{L;j}}(x)}{(x - SK'_{^a_{L;j};i})}; \tag{4}$$

turns out zero at $x = SK'_{^a_{L;j};t}$, for $t \neq i$. Therefore, if $V_i(x)$ is multiplied by $(x - D)$ for $i = 1; 2; \ldots; N_{^a_{L;j}}$, the resulting function becomes a polynomial of order $N_{^a_{L;j}}$ again, defined to be

$$U_i(x) = (x - D)V_i(x); \tag{5}$$

where D is a dummy secret key in order to make $U_i(x)$ a polynomial of degree $N_{^a_{L;j}}$. The dummy secret key D is different from the $N_{^a_{L;j}}$ pretending secret key of $^a_{L;j}$ and is only known by CA. Notice that the value $U_i(x)$ becomes zero at $x = SK'_{^a_{L;j};k}$ for $k \neq i$ by the property of (4). The basis of our proposed scheme is to use a universal key, denoted as $SK_{^a_{L;j}}$, instead of the secret keys $\{SK_{^a_{L;j};k}; k = 1; 2; \ldots; N_{^a_{L;j}}\}$ of security classes in the similar IP set $^a_{L;j}$ while any security class in $^a_{L;j}$ is constructing the interpolating

polynomial for the shared IS set $'_{L;j}$. That is, each security class in $^{a}_{L;j}$ will construct the identical interpolating polynomial for the shared IS set on the $N_{'_{L;j}} + 1$ points: $(0; SK_{a_{L;j}})$ and the $N_{'_{L;j}}$ public-parameter pairs of $'_{L;j}$ over GF(P). Now let's consider the following $N_{a_{L;j}}$ linear congruence equations:

$$SK_{a_{L;j}} = \alpha_i U_i(SK'_{a_{L;j};i}) \pmod{P}; \qquad (6)$$

for $i = 1; 2; \ldots; N_{a_{L;j}}$ where α_i's are unknown and $SK_{a_{L;j}}$ is the universal key selected by CA. Note that by the theorem 1.4 of [10], the $N_{a_{L;j}}$ linear congruence equations shown above have exactly $N_{a_{L;j}}$ solutions. Accordingly, after solving the unknown coefficients α_i's, we can have the generation polynomial for the universal key $SK_{a_{L;j}}$; given by

$$G_{a_{L;j}}(x) = \sum_{i=1}^{N_{a_{L;j}}} \alpha_i U_i(x): \qquad (7)$$

From (6) and (7), and the property of (4), we find that any security class of the similar IP set $^{a}_{L;j}$ can get the universal key $SK_{a_{L;j}}$ merely by his own corresponding pretending secret key SK', that is

$$SK_{a_{L;j}} = G_{a_{L;j}}(SK'_{a_{L;j};k}) \pmod{P}; \qquad (8)$$

for $k = 1; 2; \ldots; N_{a_{L;j}}$.

Therefore each security class in the similar IP set $^{a}_{L;j}$ can construct the identical interpolating polynomial for the shared IS set $'_{L;j}$ by the universal key $SK_{a_{L;j}}$ and the $N_{'_{L;j}}$ public-parameter pairs of $'_{L;j}$. Notice that, any security class in $^{a}_{L;j}$ can use neither the derived secret keys of the shared IS set $'_{L;j}$ nor the generation polynomial $G_{a_{L;j}}(x)$ to break the secret keys of the other peers in $^{a}_{L;j}$.

3.2 THE KEY-GENERATION ALGORITHM

The key-generation algorithm is proceeded level by level. For any security-clearance level, the security classes on the same level are categorized into similar IPs and exclusive IPs, and they are done separately by different algorithms. Accordingly, while we construct interpolating polynomials, the ISs of a node C_i in the user hierarchy are classified into two parts, if exists. One part is the shared IS set corresponding to the similar IP set to which C_i belongs, and the other part is the exclusive IS set whose exclusive IP is C_i. For any similar IP set, all IPs in this set use the corresponding universal key instead of their secret keys, while constructing interpolating polynomial for the associated shared IS set. The universal key is obtained by solving the generation polynomial in (7), which is produced via equations (3)-(6). Therefore each IP in the similar IP set can construct the identical interpolating polynomial of the corresponding shared IS set by the universal key and the public-parameter pairs of the shared IS set. As for the exclusive IPs on a security-clearance level, each of them constructs the interpolating polynomial of the associated exclusive IS set by his own secret key and the public-parameter pairs of all his exclusive ISs. Since there are two types of IPs, the proposed key-generation algorithm includes two sub-algorithms in contrast: exclusive-IP algorithm and similar-IP algorithm. The former is used for the exclusive IPs and the latter is applied on the IPs in a similar IP set. In the key-generation procedure, Step 2 to Step 4 are designed for the exclusive IPs, and Step 5 and Step 6 are applied to similar IPs. In the following, the key-generation algorithm is presented and the two sub-algorithms are shown subsequently.

Key-Generation Algorithm

Step 1:
(1a) Make all nodes in the user hierarchy unmarked.
(1b) Let L be the security-level index and set L = 1 (the highest security clearance).
Step 2:
(2a) Take an unmarked node C_i from the security classes which belongs to the Lth security clearance.
(2b) Mark C_i.
Step 3:
(3a) Determine the exclusive IS set of C_i and denote it as α_i.
(3b) Go to the exclusive-IP algorithm.
Step 4:
Repeat Step 2 and Step 3 until all nodes in the Lth security-clearance are marked.
Step 5:
(5a) Determine all the similar IP sets of the Lth security-clearance level, shown as $^{a}_{L} = \{^{a}_{L;1}; ^{a}_{L;2}; \ldots; ^{a}_{L;Q_L}\}$; and the corresponding shared IS sets, shown as $'_{L} = \{'_{L;1}; '_{L;2}; \ldots; '_{L;Q_L}\}$.
(5b) Let j be the index for the similar IP sets and default j = 1.
Step 6:
(6a) Run the similar-IP algorithm for $^{a}_{L;j}$, the jth similar set of $^{a}_{L}$.
(6b) Set j = j + 1: If $j \leq Q_L$, then return to (6a).
Step 7: If all the nodes in the user hierarchy are marked, then stop; else set L = L + 1 and return to Step2. ¥

Exclusive-IP Algorithm

Step 1:
(1a) If C_i is the root node, C_i has no IPs. Randomly select an integer SK_i between 1 and P − 1 to be the secret key of C_i. Otherwise, the secret key SK_i of C_i

has already assigned.
(1b) Suppose C_i has $N_{\Box_i}$ exclusive ISs. Randomly select $N_{\Box_i}$ distinct integers $P1_{i;1}; P1_{i;2}; :::; P1_{i;N_{\Box_i}}$ between 1 and $P - 1$, and any $N_{\Box_i}$ integers $P2_{i;1}; P2_{i;2}; :::; P2_{i;N_{\Box_i}}$ between 1 and $P - 1$.
(1c) Assign $(P1_{i;k}; P2_{i;k})$ as the public-parameter pair of the kth exclusive IS of C_i, where $k = 1; 2; :::; N_{\Box_i}$.
Step 2:
Using the Newton's interpolation method, we can construct an interpolating polynomial $H_i(x)$ of degree $N_{\Box_i}$ by interpolating on the points: $(0; SK_i^0)$ and $(P1_{i;k}; P2_{i;k})$, $k = 1; 2; :::; N_{\Box_i}$, over GF(P), shown as

$$H_i(x) = SK_i^0 + a_1x + a_2x^2 + ::: + a_{N_{\Box_i}}x^{N_{\Box_i}} \pmod P.$$

Step 3:
Compute all the secret keys for the $N_{\Box_i}$ exclusive ISs of C_i as follows.

$$SK_{i;k} = f(a_k) \pmod P; \text{ for } k = 1; 2; :::; N_{\Box_i};$$

where $SK_{i;k}$ denotes the secret key of the kth exclusive IS of C_i, and a_k is the coefficient of the term x^k in $H_i(x)$. ¥

Similar-IP Algorithm

As previously, we use $\{SK_{a_{L;j};k};\ k = 1; 2; :::; N_{a_{L;j}}\}$ to denote the $N_{a_{L;j}}$ secret keys of the jth similar IP set $a_{L;j}$ in the Lth security clearance.
Step 1:
Generate the following polynomial

$$f_{a_{L;j}}(x) = \prod_{k=1}^{N_{a_{L;j}}} (x - SK^0_{a_{L;j};k});$$

and let

$$V_i(x) = \frac{f_{a_{L;j}}(x)}{(x - SK^0_{a_{L;j};i})}; \quad \text{for } i = 1; 2; :::; N_{a_{L;j}}:$$

Step 2:
Make polynomials of degree $N_{a_{L;j}}$ in terms of $V_i(x)$:

$$U_i(x) = (x - D)V_i(x), \quad \text{for } i = 1; 2; :::; N_{a_{L;j}}:$$

where D is a dummy secret key only known by CA.
Step 3:
Set

$$SK_{a_{L;j}} = \alpha_i U_i(SK^0_{a_{L;j};i}) \pmod P;$$

for $i = 1; 2; :::; N_{a_{L;j}}$ where $SK_{a_{L;j}}$ is the predetermined universal key by CA.
Step 4:
(4a) Define a generation polynomial $G_{a_{L;j}}(x)$ for the universal key of the set $a_{L;j}$ as follows:

$$G_{a_{L;j}}(x) = \sum_{i=1}^{N_{a_{L;j}}} \alpha_i f_{a_{L;j}}(x);$$

where α_i's are obtained from Step 3.
(4b) Make $G_{a_{L;j}}(x)$ public.
Step 5:
(5a) Randomly select $N_{'_{L;j}}$ distinct integers $P1_{'_{L;j};1}; P1_{'_{L;j};2}; :::; P1_{'_{L;j};N_{'_{L;j}}}$ between 1 and $P - 1$, and any $N_{'_{L;j}}$ integers $P2_{'_{L;j};1}; P2_{'_{L;j};2}; :::; P2_{'_{L;j};N_{'_{L;j}}}$ between 1 and $P - 1$.
(5b) Assign $(P1_{'_{L;j};k}; P2_{'_{L;j};k})$ as the public-parameter pair of the kth shared IS in $'_{L;j}$, where $k = 1; 2; :::; N_{'_{L;j}}$. Note that the $N_{'_{L;j}}$ shared ISs in $'_{L;j}$ is corresponding to the jth similar IP set $a_{L;j}$.
Step 6:
Using the Newton's interpolation method, we can construct an interpolating polynomial $H_{a_{L;j}}(x)$ of degree $N_{'_{L;j}}$ by interpolating on the points: $(0; SK^0_{a_{L;j}})$ and the $N_{'_{L;j}}$ points $(P1_{'_{L;j};k};\ P2_{'_{L;j};k})$, $k = 1; 2; :::; N_{'_{L;j}}$ over GF(P), shown as

$$H_{a_{L;j}}(x) = SK^0_{a_{L;j}} + a_1x + ::: + a_{N_{'_{L;j}}}x^{N_{'_{L;j}}} \pmod P;$$

where $SK^0_{a_{L;j}} = f(SK_{a_{L;j}})$.
Step 7:
Compute all the secret keys of the shared IS set $'_{L;j}$ by

$$SK_{'_{L;j};k} = f(a_k); \quad \text{for } k = 1; 2; :::; N_{'_{L;j}};$$

where a_k is the coefficient of the term x^k in $H_{a_{L;j}}(x)$.¥

3.3 KEY-DERIVATION ALGORITHM

Assume that a security class C_i with the secret key SK_i wants to derive the secret key $SK_{i;k}$ of his IS $C_{i;k}$. As previously, the IS $C_{i;k}$ may be a node of the shared IS set corresponding to the similar IP set to which C_i belongs, or a node of the exclusive IS set whose exclusive IP is C_i. The algorithm for the key derivation is given as follows.

Key-Derivation Algorithm

Step 1:
If the security class C_i is the exclusive predecessor of $C_{i;k}$, then go to Step 2; otherwise, go to Step 3.
Step 2:
(2a) Determine the exclusive IS set $\Box_i$ of C_i and take all the corresponding public-parameter pairs of $\Box_i$, denoted as $(P1_{i;t}; P2_{i;t})$, $t = 1; 2; :::; N_{\Box_i}$, where $N_{\Box_i}$ means the cardinal number of the set $\Box_i$.
(2b) Using the Newton's interpolation method, we can

reconstruct the interpolating polynomial

$$H_i(x) = SK_i^0 + a_1x + a_2x^2 + \cdots + a_{N_{\alpha i}}x^{N_{\alpha i}} \pmod P;$$

by interpolating on the points: $(0; SK_i^0)$ and the $N_{\alpha i}$ public-parameter pairs, $(P1_{i;t}; P2_{i;t})$, $t = 1; 2; \ldots; N_{\alpha i}$, over GF(P).
(2c) Go to Step 5.
Step 3:
(3a) Determine the corresponding similar IP set ^{a} and shared IS set $'$ to which C_i and $C_{i;k}$ belongs, respectively, and then get the generation polynomial $G_a(x)$ for the universal key of the similar IP set ^{a}.
(3b) The universal key SK_a is obtained by

$$G_a(SK_i^0);$$

where SK_i^0 is the corresponding pretending secret key of C_i.
Step 4:
(4a) Take the $N_{'}$ public-parameter pairs of $'$, denoted as $(P1_{i;1}; P2_{i;1})$, $(P1_{i;2}; P2_{i;3})$; $\ldots$; $(P1_{i;N_{'}}; P2_{i;N_{'}})$.
(4b) Using the Newton's interpolation method, we can reconstruct the interpolating polynomial

$$H_i(x) = SK_i^0 + a_1x + a_2x^2 + \cdots + a_{N_{'}}x^{N_{'}} \pmod P;$$

by interpolating on the points: $(0; SK_a^0)$ and the $N_{'}$ public-parameter pairs, $(P1_{i;t}; P2_{i;t})$, $t = 1; 2; \ldots; N_{'}$, over GF(P).
Step 5:
Compute the secret key of $C_{i;k}$ by

$$SK_{i;k} = f(a_k) \pmod P;$$

where a_k is the coefficient of the term x^k of $H_i(x)$.¥

Note that the security class C_i can derive all secret keys of his successors, which could be not an immediate one, by performing the Key-Derivation Algorithm iteratively. The weakness of the original CHW scheme [8] is that it can not avoid from collaborative attack from ISs. Therefore, we substitute a corresponding pretending secret key SK^0 for its original SK for any predecessor when constructing the interpolating polynomial. Thus, we can intensify the security because even all the ISs unite together to attack the corresponding IP, and they can get nothing but a fake secret key.

4. EXAMPLES

In this section, the key-generation and key-derivation examples are given under the user hierarchy in Figure 2. There are four security-clearance levels containing twelve security classes in this user hierarchy. We suppose that the prime number P = 31 and the prede¯ned one-way function $f(x) = 7^x$. There is a CA for generating the secret key and public parameters for each security class in the user hierarchy. The generated parameters for the user hierarchy in Figure 2 are summarized at Table 2.

Key-Generation Example

² For the root node C_1
{ Randomly select the secret key $SK_1 = 7$; and (3; 12) and (10; 9) as the public-parameter pairs for C_2 and C_3, respectively.
{ Construct the interpolating polynomial $H_1(x)$ over GF(31) on the points: $(0; SK_1^0 = 28)$, (3; 12) and (10; 9); given by

$$H_1(x) = 28 + 27x + 3x^2 \pmod{31}.$$

{ Then the secret keys for C_2 and C_3 are computed as

$$SK_2 = f(27) \pmod{31} = 16 \quad \text{and}$$
$$SK_3 = f(3) \pmod{31} = 2:$$

² For exclusive IP C_2
{ The exclusive ISs for C_2 are C_4 and C_5.
{ Randomly select (15; 2) and (11; 9) as the public-parameter pairs for C_4 and C_5, respectively.
{ Construct the interpolating polynomial $H_2(x)$ over GF(31) on the points: $(0; SK_2^0 = 7)$, (15; 2) and (11; 9); given by

$$H_2(x) = 7 + 7x + 25x^2 \pmod{31}.$$

{ Then the secret keys for C_4 and C_5 are computed as

$$SK_4 = f(7) \pmod{31} = 28 \quad \text{and}$$
$$SK_5 = f(25) \pmod{31} = 25:$$

² For exclusive IP C_3
{ The exclusive ISs for C_3 are C_8 and C_9.
{ Randomly select (5; 2) and (13; 3) as the public-parameter pairs for C_8 and C_9, respectively.
{ Construct the interpolating polynomial $H_3(x)$ over GF(31) on the points: $(0; SK_3^0 = 18)$, (5; 2) and (13; 3); given by

$$H_3(x) = 18 + 5x + 12x^2 \pmod{31}.$$

{ Then the secret keys for C_8 and C_9 are computed as

$$SK_8 = f(5) \pmod{31} = 5 \quad \text{and}$$
$$SK_9 = f(12) \pmod{31} = 16:$$

² For similar IP $^a{}_{2;1} = \{C_2; C_3\}$
{ The shared ISs for $^a{}_{2;1}$ are C_6 and C_7.

{ The generation polynomial, with the dummy key D = 17, for the universal key of $a_{2;1}$ is shown as

$$G_{a_{2;1}}(x) = 6x^2 + 5x + 10;$$

for which the universal key $SK_{a_{2;1}}$ is computed as

$$SK_{a_{2;1}} = G_{a_{2;1}}(SK_2^0 = 7) = G_{a_{2;1}}(SK_3^0 = 18) = 29:$$

{ Randomly select (25; 17) and (29; 19) as the public-parameter pairs for C_6 and C_7, respectively.
{ Construct the interpolating polynomial $H_{a_{2;1}}(x)$ over GF(31) on the points: $(0; SK_{a_{2;1}}^0 = 9)$, (25; 17) and (29; 19); given by

$$H_{a_{2;1}}(x) = 9 + 19x + 12x^2:$$

{ Then the secret keys for C_6 and C_7 are computed by

$$SK_6 = f(19) \pmod{31} = 14 \quad \text{and}$$
$$SK_7 = f(12) \pmod{31} = 16:$$

² For exclusive IP C_4
{ The exclusive ISs for C_4 are $C_{10}; C_{11}$ and C_{12}.
{ Randomly select (14; 12); (7; 22) and (4; 21) as the public-parameter pairs for $C_{10}; C_{11}$ and C_{12}, respectively.
{ Construct the interpolating polynomial $H_4(x)$ over GF(31) on the points: $(0; SK_4^0 = 19)$, (14; 12); (7; 22) and (4; 21); given by

$$H_4(x) = 19 + 9x + 18x^2 + 25x^3 \pmod{31}.$$

{ Then the secret keys for $C_{10}; C_{11}$ and C_{12} are given as

$$SK_{10} = f(9) \pmod{31} = 8:$$
$$SK_{11} = f(18) \pmod{31} = 2$$
$$SK_{12} = f(25) \pmod{31} = 25:$$

² Do nothing for the leaf nodes: $C_5; C_6; \ldots;$ and C_9.

Key-Derivation Example

Suppose that C_1 wants to access the data of C_6 and C_{10}: The security class C_6 is the shared IS of $a_{2;1} = \{C_2; C_3\}$: The security class C_{10} is an immediate successor of C_4.
² The derivation of SK_6
{ Reconstruct the interpolating polynomial $H_1(x)$ over GF(31) on the points: $(0; SK_1^0 = 28)$, (3; 12) and (10; 9); given by $H_1(x) = 28 + 27x + 3x^2 \pmod{31}$:
{ The secret key of C_2 is computed by $SK_2 = f(27) \pmod{31} = 16$.
{ Get the universal key $SK_{a_{2;1}}$ for C_2 and C_3 by $SK_{a_{2;1}} = G_{a_{2;1}}(SK_2^0 = 7) = 29$:
{ Reconstruct the interpolating polynomial $H_{a_{2;1}}(x)$ over GF(31) on the points: $(0; SK_{a_{2;1}}^0 = 9)$, (25; 17) and (29; 19); given by $H_{a_{2;1}}(x) = 9 + 19x + 12x^2$:
{ The secret keys for C_6 is given by $SK_6 = f(19) \pmod{31} = 14$:

² The derivation of SK_{10}
{ Reconstruct the interpolating polynomial $H_1(x)$ over GF(31) on the points: $(0; SK_1^0 = 28)$, (3; 12) and (10; 9); given by $H_1(x) = 28 + 27x + 3x^2 \pmod{31}$:
{ The secret key of C_2 is computed by $SK_2 = f(27) \pmod{31} = 16$.
{ Reconstruct the interpolating polynomial $H_2(x)$ for the exclusive ISs of C_2 over GF(31) on the points: $(0; SK_2^0 = 7)$, (15; 2) and (11; 9); given by $H_2(x) = 7 + 7x + 25x^2 \pmod{31}$.
{ The secret key of C_4 is computed by $SK_4 = f(7) \pmod{31} = 28$.
{ Reconstruct the interpolating polynomial $H_4(x)$ for the exclusive ISs of C_4 over GF(31) on the points: $(0; SK_4^0 = 19)$, (14; 12); (7; 22) and (4; 21); given by $H_4(x) = 19 + 9x + 18x^2 + 25x^3 \pmod{31}$:
{ Then the secret key of C_{10} is computed by $SK_{10} = f(9) \pmod{31} = 8$:

5. CONCLUSIONS

A simple and e®ective scheme, based on the combination of Lagrange polynomial and Newton interpolation method, is proposed to solve the incorrectness of CHW scheme and to enhance its security at the same time. The polynomial for generating the universal key of a similar IP set is easily obtained by just solving linear congruence equations. This scheme ensures not only each security class in a similar IP set can derive each secret key of the associated shared ISs via his own secret key, without the help of his peers, but also the predecessor's secret key cannot be revealed by conspiracy of his successors.

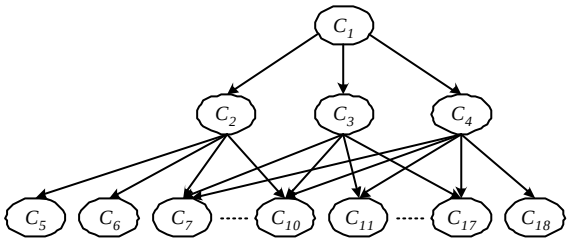

Figure 1: The poset in a user hierarchy.

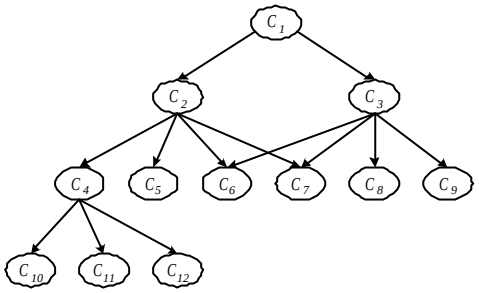

Figure 2: Examples

Table 1: The second security-clearance level of the user hierarchy in Figure 1.

Similar IP sets $^{a}{}_2$	$^{a}{}_{2;1} = \{C_2; C_3; C_4\}$
	$^{a}{}_{2;2} = \{C_3; C_4\}$
Shared IS sets $'_2$	$'_{2;1} = \{C_7; \ldots; C_{10}\}$
	$'_{2;2} = \{C_{11}; \ldots; C_{17}\}$
Immediate Successors	$©_2 = \{C_5; \ldots; C_{10}\}$
	$©_3 = \{C_7; \ldots; C_{17}\}$
	$©_4 = \{C_7; \ldots; C_{18}\}$
Exclusive IS Sets	$¤_2 = \{C_5; C_6\}$
	$¤_3 =$ Á for C_3.
	$¤_4 = \{C_{18}\}$ for C_4.

Table 2: Parameters for the user hierarchy in Figure 2.

	Security class											
	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
SK_i	7	16	2	28	25	14	16	5	16	8	2	25
SK_i^0	28	7	18	19	25	9	7	5	7	10	18	25
$P1_i$	¢¢¢	3	10	15	11	25	29	5	13	14	7	4
$P2_i$	¢¢¢	12	9	2	9	17	19	2	3	12	22	21

References

[1] S. G. Akl and P. D. Taylor, "Cryptographic solution to a problem of access control in a hierarchy," ACM Trans. Comput. Syst. Vol. 1, No. 3, pp.239-249, 1983.

[2] W. P. Lu and M. K. Sundareshan, "A model for multilevel security in computer networks," Proceedings of INFOCOM, pp.1095-1104, 1988.

[3] D. McCullough, "Speci¯cations for multilevel security and a hook-up property," Proceedings of IEEE Symposium on Security and Privacy, pp.161-166, 1987.

[4] C. C. Chang, R. J. Hwang, and T. C. Wu, "Cryptographic key assignment scheme for access control in a hierarchy," Inf. Syst., Vol. 17, No. 3, pp.243-247, 1992.

[5] S. F. 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. Comput. , Vol. 34, No. 9, pp.797-802, 1985.

[6] K. J. Tan, S. J. Gu, and H. W. Zhu, "Correctness of CHW cryptographic key assignment scheme in a hierarchy," IEE Proc. Comput. Digit. Tech., Vol. 146, No. 4, pp.217-218, 1999.

[7] M. S. Hwang, "Extension of CHW cryptographic key assignment scheme in a hierarchy," IEE Proc. Comput. Digit. Tech., Vol. 146, No. 4, p.219, 1999.

[8] M. S. Hwang, W. P. Yang, and C. C. Chang, "Modi¯ed Chang-Hwang-Wu access control scheme," Electron. Lett., Vol. 29, No. 24, pp.2095-2096, 1993.

[9] D. Knuth, The Art of Computer Programming, Vol. 2: Seminumerical Algorithm, Addison-Wesley, Reading, MA, 1969.

[10] H. E. Rose, A Course in Number Theory, p.34, Oxford, 1994.

[11] S. Nakamura, Applied Numerical Methods in C, Prentice-Hall, 1993.