Joint Conference of 1996 International Computer Symposium
December 19~21, Kaohsiung, Taiwan, R.O.C.

On the Security of Chang et al.'s Cryptographic Key Assignment Scheme for Access Control in a Hierarchy

Tzonelih Hwang[1], *Narn – Yih Lee*[2] and *Chin – Chen Chang*[3]

[1]Institute of Information Engineering, National Cheng-Kung University, Tainan

[2]Department of Applied Foreign Language, Nan-Tai Institute of Technology, Tainan

[3]Institute of Computer Science and Information Engineering, National Chung-Cheng University, Chiayi, Taiwan, R.O.C.

Abstract

In 1992, Chang, Hwang and Wu proposed a cryptographic key assignment scheme based on Newton's interpolation method and a predefined one-way function to solve the access control problem in a user hierarchy. In this paper, we shall show that their scheme is problematic in controlling access to all types of hierarchical organizations. Furthermore, we show that their scheme is not secure enough by presenting an attack on it.

1. Introduction

In the modern society, hierarchical structure of users exists in many organizations such as military and government departments or the business corporations. How to control access of data in such an environment is extremely important.

To describe this problem, let us consider a hierarchical organization in which the groups of users are denoted as disjoint classes represented by a set, U=$\{U_1, U_2, ..., U_n\}$. Define on U the relation "$\leq$" such that $U_j \leq U_i$ for some i, j means users in U_i can access any information held by users in U_j, while the opposite is not allowed (note that $U_i \leq U_i$). Such a hierarchical structure can be well modeled by an algebraic system called the *partial order set* (or called *poset*). Let U_1 be the Central Authority (**CA** in short), who is in charge of the key generation for the hierarchical organization and $U_i \leq U_1$ for all U_i in U. In the hierarchy, if $U_j \leq U_i$, then U_i is said to be a *predecessor* of U_j and U_j is said to be a *successor* of U_i. Furthermore, if there is no other class U_k in the *poset* U such that $U_j \leq U_k \leq U_i$, then U_i is called an *immediate predecessor* of U_j and U_j is called an *immediate successor* of U_i. Fig. 1 shows an example of user hierarchy with n=6.

Several schemes [1,2,4-8] have been proposed to solved the access control in a hierarchy. For examples, Akl and Taylor [4] proposed the first scheme in 1982; MacKinnon et al. [5] proposed a chain decomposition method to solve this problem. Sandhu [7] proposed a method for the special case of a tree hierarchy by employing the one-way function and ID-based concept. Harn and Lin [6] used the concept of RSA to solve the access control problem in a hierarchy. Laih and Hwang [2] proposed a branched oriented method to solve the access control problem in a hierarchy. In 1991, Zheng, Hardjono and Pieprzyk [8] proposed a scheme to solve this problem based on the *sibling intractable function families*.

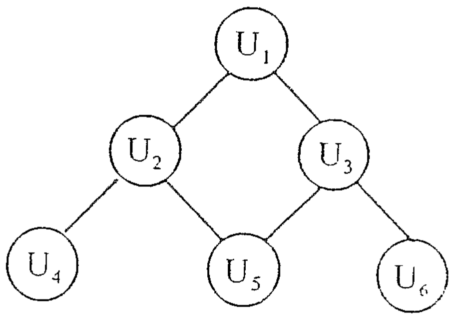

Figure 1. A user hierarchy system

Recently, Chang, Hwang and Wu [1] proposed a scheme based on *Newton's interpolation method* and a predefined one-way function to solve the access control problem. In this paper, we are going to propose two problems with Chang et al.'s scheme in controlling accesses for all types of user hierarchies in Section 3. Then, we shall propose an attack to their scheme in Section 4. In Section 5, a concluding remark will be given. The original Chang et al.'s scheme is given in the next section.

2. Review of Chang et al.'s scheme

In their scheme, the CA has to generate both the secret key K_i and a pair of public parameters $(s_i,\ t_i)$ for each user class U_i. Each user class U_i can use his secret key K_i to encrypt (decrypt) its files, and also by some computations on U_i's secret key and the public parameters of U_i's *immediate successors*, U_i can derive the secret keys of its *immediate successors*. Here, we use the user hierarchy shown in Fig. 1 to describe Chang et al.'s *Key Generation Algorithm*.

[Chang et al.'s Key Generation Algorithm]

Step 1 : Unmark all nodes in the hierarchy.

Step 2 : Get an unmarked node U_i from the hierarchy by **inorder** traversal.

Step 3 : If U_i is a leaf node, then go to *Step* 8.

Step 4 : Let k be the number of U_i's immediate successors, and U_{i1}, U_{i2}, ..., U_{im} be the unmarked immediate successors and $U_{i,m+1}$, $U_{i,m+2}$, ..., U_{ik} be the marked immediate successors.

Step 5 : If U_i is the root node, then do the following :

(1) Randomly choose a key K_i for U_i, where $1 \leq K_i \leq$ P-1, P is a large prime.

(2) Mark U_i.

(3) Randomly choose a polynomial of degree k in $GF(\text{P})$ with K_i as the constant term. The polynomial is denoted as

$$H_i(X) = K_i + a_1X + a_2X^2 + \ \dots\ + a_kX^k \pmod{\text{P}}$$

where $a_1, a_2,\ \dots,\ a_k$ are integers between 1 and P-1.

(4) Randomly choose k distinct integers s_{i1}, s_{i2}, ..., s_{ik} between 1 and P-1 and compute

$t_{ij} = H_i(s_{ij})$, for $1 \leq j \leq k$.

(s_{ij}, t_{ij}) are the public parameters of U_{ij}. Go to *Step* 7.

Step 6 : If U_i is not the root node, i.e. the secret key of U_i has already been assigned, then do the following :

(1) Each U_i's marked immediate successor U_{il} has a pair of public parameters denoted as (s_{il}, t_{il}) for $m+1 \leq l \leq k$.

(2) Randomly choose m distinct integers s_{i1}, s_{i2}, ..., s_{im} between 1 and P-1 such that $s_{ij} \neq s_{il}$ for $1 \leq j \leq m$ and $m+1 \leq l \leq k$.

(3) Randomly choose m integers t_{ij} between 1 and P-1 for $1 \leq j \leq m$.

(4) Use the *Newton's interpolation method* to construct the interpolating polynomial $H_i(X)$ of degree k that passes the points $(0, K_i)$, (s_{i1}, t_{i1}), (s_{i2}, t_{i2}), ..., (s_{ik}, t_{ik}). Let the polynomial be denoted as

$$H_i(X) = K_i + a_1X + a_2X^2 + \ldots + a_kX^k \pmod{P}$$

Step 7 : Compute the secret key K_{ij} of U_i's *immediate successor* U_{ij} by

$$K_{ij} = f(a_j) \pmod{P}, \quad \text{for } 1 \leq j \leq m.$$

where a_j is the coefficient of the term X^j in $H_i(X)$ and f is a predefined one-way function.

Mark U_{ij}, for $1 \leq j \leq m$.

Step 8: Repeat *Step* 2 until all nodes in the hierarchy are marked.

After the *Key Generation Algorithm*, CA assigns each user class U_i both a secret key K_i and a pair of public parameters (s_i, t_i). Thus, if the user class U_i wants to derive its *immediate successor* U_j's secret key K_j, then U_i can use his secret key K_i and the public parameters of all its *immediate successors* to reconstruct the interpolating polynomial $H_i(X)$. Let U_i have k *immediate successors*, and the interpolating polynomial $H_i(X)$ be represented as follows.

$$H_i(X) = a_o + a_1X + a_2X^2 + \ldots + a_kX^k \pmod{P},$$

where a_i is the coefficient of the term X^i.

Thus, U_i can compute the secret key K_j of U_j by

$$K_j = f(a_j) \pmod{P}.$$

Note that if U_j is not a U_i's immediate successor, then U_i has to compute the key of its immediate successors U_k and then the key of U_k's immediate successor step by step until obtaining the key of U_j.

3. On the correctness of the algorithm

Chang et al.'s algorithm traverses the hierarchical tree in Fig. 1 by **inorder** traversal. Therefore, the sequence, on which the nodes of the tree were visited, is U_4, U_2, U_5, U_1, U_3, and U_6. It is obvious that U_2 will be visited before the root node U_1. However, U_2 still has not been assigned a secret key yet. Thus the algorithm fails at *Step* 2. If the algorithm is modified to assign the U_2 a secret key at *Step* 6, then the algorithm will have trouble in assigning a unique secret key to each of the *immediate successors* of the root node. In other words, some nodes (e.g., U_2) will have two secret keys. Therefore, the root node should be traversed first before the other nodes, i.e., the tree has to be traversed in **preorder** traversal.

In addition to the problem described above, we are going to show that Chang et al.'s algorithm cannot always construct *Newton interpolating polynomials* for all types of user hierarchies.

Theorem 1: Given $(m+1)$ coefficients of a *Newton interpolating polynomial* $H(X)$ of degree n in $GF(\mathrm{P})$ and m distinct points, where $2m > n$, the probability for $H(X)$ to visit these m given points is $\frac{1}{P^{2m-n}}$.

(Proof)

Assume that the polynomial $H(X)$ given above is denoted as

$$H(X) = a_o + a_1X + a_2X^2 + \ldots + a_mX^m + b_1X^{m+1} + b_2X^{m+2} + \ldots + b_{n-m}X^n \pmod{P},$$

where $a_i's$, $1 \le i \le n-m$, are the given coefficients and $b_i's$, $1 \le j \le n-m$, are unknowns.

Let these m distinct points, which have to be visited by $H(X)$, be denoted as (s_1, t_1), (s_2, t_2), ..., (s_m, t_m) where s_i, $t_i \in GF(\mathrm{P})$, $1 \le i \le m$. Then, we have the following $(n-m)$ equations : ($H(X)$ visits the first $(n-m)$ points)

$$\begin{cases} t_1 = a_o + a_1s_1 + a_2s_1^2 + \ldots + a_ms_1^m \\ \quad + b_1s_1^{m+1} + b_2s_1^{m+2} + \ldots + b_{n-m}s_1^n \\ \quad \pmod{P}, \\ t_2 = a_o + a_1s_2 + a_2s_2^2 + \ldots + a_ms_2^m \\ \quad + b_1s_2^{m+1} + b_2s_2^{m+2} + \ldots + b_{n-m}s_2^n \\ \quad \pmod{P}, \\ \vdots \\ t_{n-m} = a_o + a_1s_{n-m} + a_2s_{n-m}^2 + \ldots + \\ \quad a_ms_{n-m}^m + b_1s_{n-m}^{m+1} + b_2s_{n-m}^{m+2} + \ldots \\ \quad + b_{n-m}s_{n-m}^n \pmod{P}. \end{cases}$$

These equations can be rewritten as follows.

$$\begin{cases} b_1 + b_2s_1 + \ldots + b_{n-m}s_1^{n-m-1} = A_1 \\ \quad \pmod{P}, \\ b_1 + b_2s_2 + \ldots + b_{n-m}s_2^{n-m-1} = A_2 \\ \quad \pmod{P}, \\ \vdots \\ b_1 + b_2s_{n-m} + \ldots + b_{n-m}s_{n-m}^{n-m-1} = A_{n-m} \\ \quad \pmod{P}, \end{cases}$$

where $A_i = s_i^{-m-1}(t_i - a_o - a_1s_i - a_2s_i^2 - \ldots - a_ms_i^m) \pmod{P}$. That is :

$$\begin{bmatrix} 1 & s_1 & s_1^2 & \ldots & s_1^{n-m-1} \\ 1 & s_2 & s_2^2 & \ldots & s_2^{n-m-1} \\ & \vdots & & \ddots & \vdots \\ 1 & s_{n-m} & s_{n-m}^2 & \ldots & s_{n-m}^{n-m-1} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-m} \end{bmatrix} = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_{n-m} \end{bmatrix} \quad (1)$$

Since the matrix

$$S = \begin{bmatrix} 1 & s_1 & s_1^2 & \ldots & s_1^{n-m-1} \\ 1 & s_2 & s_2^2 & \ldots & s_2^{n-m-1} \\ & \vdots & & \ddots & \vdots \\ 1 & s_{n-m} & s_{n-m}^2 & \ldots & s_{n-m}^{n-m-1} \end{bmatrix}$$

is a *Vandermonde* matrix [3], the inverse matrix S^{-1} of S can be computed.

By (1), we have

$$\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-m} \end{bmatrix} = \begin{bmatrix} S^{-1} \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_{n-m} \end{bmatrix}$$

At this point, all coefficients of the interpolating polynomial $H(X)$ have been already decided.

However, according to the theorem, there are still $2m-n$ distinct points has to be visited by $H(X)$. Since the probability of $H(X)$ to visit the point (s_i, t_i) is $\frac{1}{P}$, for $n-m+1 \leq i \leq m$, the probability of $H(X)$ to visit all these $2m-n$ points is $\frac{1}{P^{2m-n}}$.

(Q.E.D.)

The above theorem implies that Chang et al.'s algorithm has the chance of only $\frac{1}{P^{2m-n}}$ to construct a *Newton interpolating polynomial* $H_i(X)$ for U_i if U_i has n *immediate successors* with m ($m > \frac{n}{2}$) of them being marked (assigned public parameters) previously.

4. On the security of the algorithm

In the above section, we show that Chang et al.'s scheme may not be able to construct *Newton interpolating polynomials* for all types of hierarchical organizations. In this section, we shall show that their scheme cannot provide adequate security for some types of hierarchical organizations. In particular, we are going to show that if a user class $U_c(\in \mathrm{U})$ has k *immediate successors* with at least one of them shared with the other user class U_d, then U_d can compute the secret key K_c of U_c from the public parameters of U_c's k *immediate successors*, and vice versa.

Theorem 2 : Knowing one non-constant term's coefficient of an interpolating polynomial $H(X)$ of degree n in $GF(\mathrm{P})$ and n distinct points visited by $H(X)$, the interpolating polynomial $H(X)$ can be reconstructed.

⟨Proof⟩ Assume that the polynomial $H(X)$ of degree n in $GF(\mathrm{P})$ is denoted as

$$H(X) = a_o + a_1 X + a_2 X^2 + \ldots + a_n X^n \pmod{\mathrm{P}},$$

where P is a large prime number, $a_i \in GF(\mathrm{P})$, $\forall$ i, $0 \leq i \leq n$.

Let these n distinct points visited by $H(X)$ be denoted as (s_1, t_1), (s_2, t_2), ..., (s_n, t_n), where $s_i,\ t_i \in GF(\mathrm{P})$, $1 \leq i \leq n$. Then, we have the following n equations

$$\begin{cases} t_1 = a_o + a_1 s_1 + a_2 s_1^2 + \ldots + a_n s_1^n \pmod{\mathrm{P}}, \\ t_2 = a_o + a_1 s_2 + a_2 s_2^2 + \ldots + a_n s_2^n \pmod{\mathrm{P}}, \\ \vdots \\ t_n = a_o + a_1 s_n + a_2 s_n^2 + \ldots + a_n s_n^n \pmod{\mathrm{P}}, \end{cases}$$

Without loss of generality, assume that the coefficient a_k of $H(X)$, for $0 \leq k \leq n$, was known. Thus, these equations can be rewritten as follows.

$$\begin{cases} a_o + a_1 s_1 + \ldots + a_{k-1} s_1^{k-1} + a_{k+1} s_1^{k+1} + \\ \qquad \ldots + a_n s_1^n = A_1 \qquad \pmod{\mathrm{P}}, \\ a_o + a_1 s_2 + \ldots + a_{k-1} s_2^{k-1} + a_{k+1} s_2^{k+1} + \\ \qquad \ldots + a_n s_2^n = A_2 \qquad \pmod{\mathrm{P}}, \\ \qquad \vdots \\ a_o + a_1 s_n + \ldots + a_{k-1} s_n^{k-1} + a_{k+1} s_n^{k+1} + \\ \qquad \ldots + a_n s_n^n = A_n \qquad \pmod{\mathrm{P}}, \end{cases}$$

where $A_i = t_i - a_k s_i^k \pmod{P}$, for $1 \le i \le n$.

That is :

$$\begin{bmatrix} 1 & s_1 & s_1^2 & \dots & s_1^{k-1} & s_1^{k+1} & \dots & s_1^n \\ 1 & s_2 & s_2^2 & \dots & s_2^{k-1} & s_2^{k+1} & \dots & s_2^n \\ & \vdots & & \ddots & & & & \vdots \\ 1 & s_n & s_n^2 & \dots & s_n^{k-1} & s_n^{k+1} & \dots & s_n^n \end{bmatrix} \begin{bmatrix} a_o \\ a_1 \\ \vdots \\ a_{k-1} \\ a_{k+1} \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{bmatrix} \quad (2)$$

Let

$$S = \begin{bmatrix} 1 & s_1 & s_1^2 & \dots & s_1^{k-1} & s_1^{k+1} & \dots & s_1^n \\ 1 & s_2 & s_2^2 & \dots & s_2^{k-1} & s_2^{k+1} & \dots & s_2^n \\ & \vdots & & & & & \ddots & \vdots \\ 1 & s_n & s_n^2 & \dots & s_n^{k-1} & s_n^{k+1} & \dots & s_n^n \end{bmatrix}$$

S should be an n n nonsingular matrix. Otherwise, $H(X)$ cannot be uniqnely decided. Thus the inverse matrix S^{-1} of S can be computed.

By (2), we have

$$\begin{bmatrix} a_o \\ a_1 \\ \vdots \\ a_{k-1} \\ a_{k+1} \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} \\ S^{-1} \\ \\ \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{bmatrix}$$

Therefore, all coefficients of the interpolating polynomial $H(X)$ have been already decided.

(Q.E.D.)

By *Theorem* 2, if U_c shares at least one immediate successor with U_d, then U_d can reconstruct the interpolating polynomial $H_c(X)$ of U_c and then obtains the secret key K_c of U_c.

5. Conclusions

In this paper, we have shown that the cryptographic key assignment scheme proposed in [1] based on the *Newton's interpolation polynomial method* and a predefined one-way function to solve the access control problem in a hierarchy is unsound. For a particular user class in a hierarchy, if the number of its marked *immediate successors* is greater than the number of its unmarked *immediate successors*, then the key generation algorithm in [1] may not be able to construct a *Newton interpo-lating polynomial* to that user class. Furthermore, we have shown that their scheme is not secure enough because if two user classes share at least a common *immediate successor*, then one can derive the other's secret key.

Acknowledgement. The authors wish to thank the anonymous referees for their useful comments.

References

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

[2] C. S. Laih and T. Hwang, "A branch oriented key management solution to dynamic access control in a hierarchy," *1991 Symposium on Applied Computing*, Kansas City, Missouri, USA, Apr. 3-5, pp. 422-429.

[3] K. Hoffman and R. Kunze, " Linear Algebra," Second Edition, *Prentice -Hall, Englewood Cliffs, N.J.*, 1971, pp. 117-139.

[4] L. Harn and H. Y. Lin, "A cryptographic key generation scheme for multilevel data security," *Computers & Security*, 9 (1990) pp. 539-546.

[5] R. S. Sandhu, "Cryptographic implementation of a tree hierarchy for access control," *Information Processing Letter*, 27 (1988) pp. 95-98.

[6] S.G. Akl and P.D. Taylor, "Cryptographic solution to a multilevel security problem," *Proc. Crypto 82*, Santa Barbara, CA, August 23-25, 1982, pp. 237-250.

[7] S. J. MacKinnon, P. D. Taylor, H. Meijer and S. G. Akl, " An optimal algorithm for assigning cryptographic keys to access control in a hierarchy," *IEEE Transations on Computers*, C-34(9) (Semptember 1985) pp. 797-802.

[8] Y. Zheng, T. Hardjono and J. Pieprzyk, " Sibling intractable function families and their applications," *Proc. AsiaCrypto 91*, 1992, pp. 67-74.