Computers Math. Applic. Vol. 24, No. 11, pp. 71–78, 1992
Printed in Great Britain.

0097-4943/92 $5.00 + 0.00

AN EFFICIENT HIERARCHY SCHEME FOR DETERMINING RELATIONSHIPS IN INFORMATION SYSTEMS

HORNG-TWU LIAW AND CHIN-LAUNG LEI
Department of Electrical Engineering, National Taiwan University
Taipei, Taiwan 106, R.O.C.

(Received November 1991)

Abstract—In this paper, a dynamic user hierarchy mechanism based on key pairs is proposed. Through careful design of keys, we get an economic integer key pair for each user. In our scheme, the relationship between two users can be easily revealed by performing a simple formula to the key pair. Furthermore, by applying our scheme, whenever a new user is inserted into the user hierarchy system, the corresponding key pair can be determined quickly without changing any existing key pairs.

1. INTRODUCTION

Owing to the popularity of computer networks and fast progress of computer technologies on a multiuser system, tremendous information is accessed and shared among the users of the system. However, sharing may cause some undesired phenomena such as unauthorized accesses and inconsistent status of shared resources. Therefore, in a multiusers computing environment, how to control accesses to the shared resources and/or identify whether a user has enough privilege to alter the access right of another user or not are important issues. To date, many papers [1–13] have addressed this problem.

In a computer protection system, the user hierarchy mechanism is always used to represent the relationships among users. The information on these relationships can be assigned to handle the requests to change access attributes. For example, changing the read, write or execution rights, deleting old users, and inserting new users are typical requests of changing privileges. Figure 1 shows a user hierarchy structure. In that figure, we use an ID number to represent each user. In addition, each branch connecting the adjacent users represents the father-son relation between these two users.

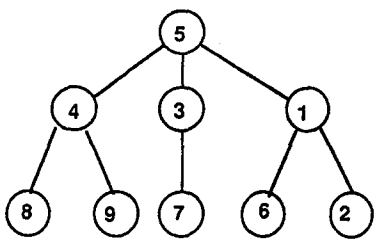


Figure 1. The user hierarchy structure.

Recently, researchers have devoted themselves to finding efficient schemes for implementing the user hierarchy structures. Many approaches have been proposed, such as the ownership hierarchy of subjects by Graham *et al.* [6], the access hierarchy of users by Gudes [7], the access control hierarchy by Saltzer *et al.* [12], the key-key pair mechanism by Wu *et al.* [13] and the user hierarchy mechanism based upon the Chinese remainder theorem by Chang [4]. In this paper, we propose a new user hierarchy mechanism which assigns each user an economic key pair, so that

This work was supported in part by the National Science Council of the Republic of China under the grant NSC 80-0416-E-002-04.

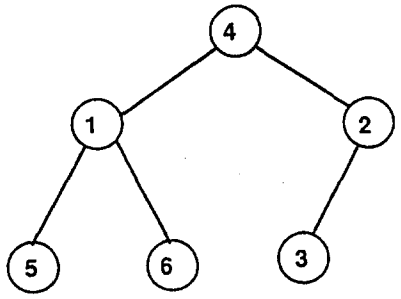
the relationship between any two users can be derived by performing a simple formula to their corresponding key pairs. The rest of this paper is organized as follows: In Section 2, we discuss the related research. Our efficient scheme for determining relationships in information systems is described in Section 3. Section 4 proves that the keys generated by our scheme are smaller than the keys obtained by [4]. Besides, the space needed by our scheme is also less than that of [13]. Finally, concluding remarks are given in Section 5.

2. A RELATED RESEARCH

Among the schemes [4,6,7,12,13] mentioned in the previous section, only the methods proposed by Wu *et al.* [13] and Chang [4] adopt the implicit methods to design. They assign each user a key and through a mathematical operation between any two keys, the relationship between the associated users is deduced. In the following, we illustrate these two methods through two simple examples, respectively.

Example 1

Consider a user hierarchy structure with six users as shown in Figure 2(a). Figure 2(b) shows the matrix representation.



(a) The user hierarchy structure.

user i \ user j	1	2	3	4	5	6
1	0	1	0	4	2	2
2	1	0	2	4	0	0
3	0	4	0	5	0	0
4	2	2	3	0	3	3
5	4	0	0	5	0	1
6	4	0	0	5	1	0

0: No relation
1: Brother of
2: Father of
3: Grandfather of
4: Son of
5: Grandson of

(b) The matrix representation.

Figure 2.

It can be seen that the matrix is in essence symmetric by close inspection of the matrix representation.

This interpretation is that the relation for User i to User j is the reverse of the relation for User j to User i. That is, if h_{ij}, the relationship of User i to User j, reveals "father of," then h_{ij}, the relationship of User j to User i, means "son of." In addition, each element in the main diagonal, which represents ones relationship to himself, is useless. Therefore, the lower triangle can be used to represent the whole matrix. This lower triangle matrix is called the matrix H, as shown in Figure 3.

user i \ user j	1	2	3	4	5	6
1	0	0	0	0	0	0
2	1	0	0	0	0	0
3	0	4	0	0	0	0
4	2	2	3	0	0	0
5	4	0	0	5	0	0
6	4	0	0	5	1	0

0: No relation
1: Brother of
2: Father of
3: Grandfather of
4: Son of
5: Grandson of

Figure 3. The matrix H.

By applying the mechanism of [13], we then construct a key matrix in order to find all the users keys. The matrix K is constructed as follows:

$$K = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} \\ k_{21} & k_{22} & k_{23} & k_{24} & k_{25} & k_{26} \\ k_{31} & k_{32} & k_{33} & k_{34} & k_{35} & k_{36} \\ k_{41} & k_{42} & k_{43} & k_{44} & k_{45} & k_{46} \\ k_{51} & k_{52} & k_{53} & k_{54} & k_{55} & k_{56} \\ k_{61} & k_{62} & k_{63} & k_{64} & k_{65} & k_{66} \end{bmatrix}, \qquad \text{such that } K_i * K_j = h_{ij},$$

where $1 \leq j < i \leq 6$, K_i represents the i^{th} row vector of K and $*$ is the inner product computation of GF(7); here, 7 is the smallest prime number larger than all elements in the matrix H and chosen as the module number of the Galois field. Then we reserve fifteen lower triangle variables and randomly assign values to the other twenty-one elements in the matrix K. Thus, the matrix K becomes

$$K = \begin{bmatrix} 2 & 4 & 0 & 3 & 1 & 0 \\ k_{21} & 2 & 0 & 0 & 0 & 0 \\ k_{31} & k_{32} & 1 & 0 & 0 & 1 \\ k_{41} & k_{42} & k_{43} & 1 & 0 & 3 \\ k_{51} & k_{52} & k_{53} & k_{54} & 1 & 0 \\ k_{61} & k_{62} & k_{63} & k_{64} & k_{65} & 1 \end{bmatrix}.$$

Furthermore, we obtain all keys by solving the fifteen equations:

$$\begin{aligned}
&k_{21} * 2 + 2 * 4 + 0 * 0 + 0 * 3 + 0 * 1 + 0 * 0 && = h_{21} = 1, \\
&k_{31} * 2 + k_{32} * 4 + 1 * 0 + 0 * 3 + 0 * 1 + 1 * 0 && = h_{31} = 0, \\
&k_{31} * k_{21} + k_{32} * 2 + 1 * 0 + 0 * 0 + 0 * 0 + 1 * 0 && = h_{32} = 4, \\
&k_{41} * 2 + k_{42} * 4 + k_{43} * 0 + 1 * 3 + 0 * 1 + 3 * 0 && = h_{41} = 2, \\
&k_{41} * k_{21} + k_{42} * 2 + k_{43} * 0 + 1 * 0 + 0 * 0 + 3 * 0 && = h_{42} = 2, \\
&k_{41} * k_{31} + k_{42} * k_{32} + k_{43} * 1 + 1 * 0 + 0 * 0 + 3 * 1 && = h_{43} = 3, \\
&k_{51} * 2 + k_{52} * 4 + k_{53} * 0 + k_{54} * 3 + 1 * 1 + 0 * 0 && = h_{51} = 4, \\
&k_{51} * k_{21} + k_{52} * 2 + k_{53} * 0 + k_{54} * 0 + 1 * 0 + 0 * 0 && = h_{52} = 0, \\
&k_{51} * k_{31} + k_{52} * k_{32} + k_{53} * 1 + k_{54} * 0 + 1 * 0 + 0 * 1 && = h_{53} = 0, \\
&k_{51} * k_{41} + k_{52} * k_{42} + k_{53} * k_{43} + k_{54} * 1 + 1 * 0 + 0 * 3 && = h_{54} = 5, \\
&k_{61} * 2 + k_{62} * 4 + k_{63} * 0 + k_{64} * 3 + k_{65} * 1 + 1 * 0 && = h_{61} = 4, \\
&k_{61} * k_{21} + k_{62} * 2 + k_{63} * 0 + k_{64} * 0 + k_{65} * 0 + 1 * 0 && = h_{62} = 0, \\
&k_{61} * k_{31} + k_{62} * k_{32} + k_{63} * 1 + k_{64} * 0 + k_{65} * 0 + 1 * 1 && = h_{63} = 0, \\
&k_{61} * k_{41} + k_{62} * k_{42} + k_{63} * k_{43} + k_{64} * 1 + k_{65} * 0 + 1 * 3 && = h_{64} = 5, \\
&k_{61} * k_{51} + k_{62} * k_{52} + k_{63} * k_{53} + k_{64} * k_{54} + k_{65} * 1 + 1 * 0 && = h_{65} = 1.
\end{aligned}$$

After the fifteen equations are solved in the Galois field GF(7), we obtain

$$K = \begin{bmatrix} 2 & 4 & 0 & 3 & 1 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 3 & 2 & 1 & 0 & 0 & 1 \\ 1 & 1 & 2 & 1 & 0 & 3 \\ 10 & 0 & 5 & 6 & 1 & 0 \\ 4 & 0 & 1 & 3 & 1 & 1 \end{bmatrix}.$$

That is, the key vectors of all users are $K_1 = (2, 4, 0, 3, 1, 0)$, $K_2 = (0, 2, 0, 0, 0, 0)$, $K_3 = (3, 2, 1, 0, 0, 1)$, $K_4 = (1, 1, 2, 1, 0, 3)$, $K_5 = (10, 0, 5, 6, 1, 0)$, and $K_6 = (4, 0, 1, 3, 1, 1)$, respectively.

Here, the determinant D of the matrix K is not zero. One thing should be noticed is, if $D = 0$, we must reassign the main diagonal and the upper triangle variables in the original matrix K again and renewedly find the remaining fifteen variables until the resulting matrix K is nonsingular.

Now, if we want to identify whether User 3 has the privilege to access the files constructed by User 2 or not, we can simply compute the expression $K_3 * K_2 = (3, 2, 1, 0, 0, 1) * (0, 2, 0, 0, 0, 0) = 4 = h_{32}$ and obtain that User 3 is a son of User 2. On the other hand, if we would like to find the relation for User 3 to User 4, we must evaluate the relation for User 4 to User 3 first, and then reverse the relation. That is, $K^4 * K^3 = (1, 1, 2, 1, 0, 3) * (3, 2, 1, 0, 0, 1) = 3 = h_{43}$ and reverse(3) = 5.

The main advantage of this mechanism is its simplicity. The relationship between two users can be derived by performing a single multiplication operation on two vectors. On the other hand, it is necessary to record the key vector of each user, that is, $O(n^2)$ space is required, where n is the total number of users.

Later, Chang [4] proposed a user hierarchy mechanism based upon the Chinese remainder theorem. In a computer protection system of n users, User i possesses a key K_i and a coprime integer q_i. H_{n*n} is a predefined matrix H and h_{ij} is the $(i,j)^{\text{th}}$ element of the matrix H. Given a finite set $S = \{q_1, q_2, \ldots, q_n\}$ of coprime integers for n users, where $q_j > \max\{h_{ij}\}_{1 \le i,j \le n}$, the relation for User i to User j is revealed by

$$h_{ij} = \begin{cases} K_i \bmod q_i, & \text{if } i > j, \\ \text{reverse } (K_j \bmod q_i), & \text{otherwise.} \end{cases}$$

In [4], Chang proposed an algorithm to compute K_i's which are smaller than $q_1 * q_2 * \cdots * q_n$. The interested readers are referred to [4] for more details.

Example 2

Let us consider the matrix H in Example 1 again. Assume $q_1 = 7$, $q_2 = 8$, $q_3 = 9$, $q_4 = 11$, $q_5 = 13$, and $q_6 = 17$. By using the Chinese remainder theorem, there exists $K_1 = 0$, $K_2 = 350064$, $K_3 = 612612$, $K_4 = 598026$, $K_5 = 4184856$, and $K_6 = 4750344$ satisfying $h_{ij} = K_i \bmod q_j$, for $1 \le j < i \le 6$.

Let us consider "h_{42}," for example. $K_4 = 598026$ and $q_2 = 8$ are selected, and then the following operation is performed:

$$h_{42} = K_4 \bmod q_2 = 598026 \bmod 8 = 2.$$

The main drawback of Chang's method is that keys grow very rapidly when the total number of users becomes large. In the next section, we present a new user hierarchy mechanism. In our method, each legal user is given a digital key pair. Through careful design of keys, we get a more economic key for each user than that of Chang's method.

3. OUR APPROACH

In this section, we present a scheme that fulfils the requirement of user hierarchy protection system. Let A_{n*n} show the corresponding matrix representation of a user hierarchy structure, where n is the number of users. Described below is a theorem that reveals any privilege value a_{ij} of A_{n*n}.

THEOREM 3.1. *Let A_{n*n} show the corresponding matrix representation of a user hierarchy structure, and A_i represent the i^{th} row vector of A. Besides, let t_i, $1 \le i \le n$, be the smallest integer larger than all elements in A_i, and a_{ij}, $1 \le i, j \le n$, represent the relationship of User i to User j. Then there exists an integer $K_i = \sum_{m=1}^{i-1} a_{im}\, t_i^{m-1}$, called the key of User i, such that*

$$a_{ij} = \begin{cases} \lfloor \frac{K_i}{t_i^{j-1}} \rfloor \bmod t_i, & \text{if } i > j, \\ \text{reverse } \left(\lfloor \frac{K_i}{t_j^{i-1}} \rfloor \bmod t_j \right), & \text{if } i < j, \\ 0, & \text{if } i = j. \end{cases} \tag{3.1}$$

PROOF. Case 1: $i > j$. Since $K_i = \sum_{m=1}^{i-1} a_{im}\, t_i^{m-1}$, we have

$$\begin{aligned}\lfloor \frac{K_i}{t_i^{j-1}} \rfloor &= \lfloor \frac{a_{i1} + a_{i2}\, t_i + \cdots + a_{i,i-1}\, t_i^{i-2}}{t_i^{j-1}} \rfloor \\ &= \lfloor a_{i1}\, t_i^{1-j} + \cdots + a_{i,j-1}\, t_i^{-1} + a_{ij} + a_{i,j+1}\, t_i + \cdots + a_{i,i-1}\, t_i^{i-j-1} \rfloor \\ &= \lfloor B + a_{ij} + C \rfloor,\end{aligned}$$

where $B = a_{i1}\, t_i^{1-j} + \cdots + a_{i,j-1}\, t_i^{-1}$ and $C = a_{i,j+1}\, t_i + \cdots + a_{i,i-1}\, t_i^{i-j-1}$.

Owing to t_i is the smallest integer larger than all elements in A_i, we have $t_i > a_{ij}$, for $1 \le i, j \le n$. Thus,

$$\begin{aligned}B &= a_{i1}\, t_i^{1-j} + \cdots + a_{i,j-1}\, t_i^{-1} \le (t_i - 1)\, t_i^{1-j} + \cdots + (t_i - 1)\, t_i^{-1} \\ &= (t_i - 1)\, \frac{t_i^{1-j}\,(1 - t_i^{j-1})}{1 - t_i} \\ &= 1 - \left(\frac{1}{t_i}\right)^{j-1}.\end{aligned}$$

Equation (3.2) implies that $0 \le B < 1$, for $1 \le i, j \le n$. Furthermore,

$$\begin{aligned}C &= a_{i,j+1}\, t_i + \cdots + a_{i,i-1}\, t_i^{i-j-1} = t_i\,(a_{i,j+1} + \cdots + a_{i,i-1}\, t_i^{i-j-2}) \\ &= t_i * R, \quad \text{where } R \text{ is an integer.}\end{aligned}$$

Therefore, we conclude that

$$a_{ij} = \lfloor \frac{K_i}{t_i^{j-1}} \rfloor \bmod t_i = \lfloor B + a_{ij} + C \rfloor \bmod t_i = a_{ij}.$$

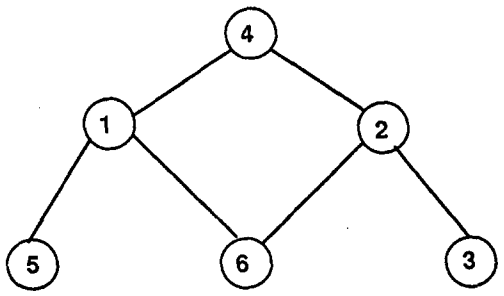
Case 2: $i < j$. Let $K_j = \sum_{m=1}^{j-1} a_{jm}\, t_j^{m-1}$. Similarly, the formula $a_{ji} = \lfloor K_j / t_j^{i-1} \rfloor \bmod t_j$ can be derived. Thus, we can easily obtain that $a_{ij} =$ reverse (a_{ji}), and this case is derived.

Case 3: $i = j$. Since each element in main diagonal, which represents one's relationship to himself, is useless, therefore, it is easy to see that this case is trivial and true. ∎

In the following, an example is given to illustrate the above theorem.

Example 3

Consider a user hierarchy structure with six users as depicted in Figure 4(a). Figure 4(b) shows the corresponding matrix A.



(a) The user hierarchy structure.

user i \ user j	1	2	3	4	5	6
1	0	1	0	4	2	2
2	1	0	2	4	0	2
3	0	4	0	5	0	1
4	2	2	3	0	3	3
5	4	0	0	5	0	1
6	4	4	1	5	1	0

0: No relation
1: Brother of
2: Father of
3: Grandfather of
4: Son of
5: Grandson of

(b) The matrix representation.

Figure 4.

By Theorem 3.1, each key can be computed as $K_i = \sum_{m=1}^{i-1} a_{im}\, t_i^{m-1}$, where $t_1 = 5$, $t_2 = 5$, $t_3 = 6$, $t_4 = 4$, $t_5 = 6$, and $t_6 = 6$. That is,

$$K_1 = 0,$$
$$K_2 = 1 * 5^0 = 1,$$
$$K_3 = 0 * 6^0 + 4 * 6^1 = 24,$$
$$K_4 = 2 * 4^0 + 2 * 4^1 + 3 * 4^2 = 2 + 8 + 48 = 58,$$
$$K_5 = 4 * 6^0 + 0 * 6^1 + 0 * 6^2 + 5 * 6^3 = 4 + 1080 = 1084, \quad \text{and}$$
$$K_6 = 4 * 6^0 + 4 * 6^1 + 1 * 6^2 + 5 * 6^3 + 1 * 6^4 = 4 + 24 + 36 + 1080 + 1296 = 2440.$$

Thus, we have six key pairs

$$(K_1, t_1) = (0, 5),$$
$$(K_2, t_2) = (1, 5),$$
$$(K_3, t_3) = (24, 6),$$
$$(K_4, t_4) = (58, 4),$$
$$(K_5, t_5) = (1084, 6), \quad \text{and}$$
$$(K_6, t_6) = (2440, 6).$$

If we want to identify whether User 3 has the privilege to access the files constructed by User 2 or not, the corresponding access control privilege is computed as

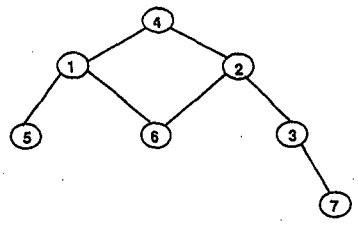
$$a_{32} = \lfloor \frac{K_3}{t_3^1} \rfloor \bmod t_3 = \lfloor \frac{24}{6^1} \rfloor \bmod 6 = 4 \bmod 6 = 4.$$

Thus, we obtain that User 3 is a son of User 2. On the other hand, if we would like to find the relation for User 3 to User 4, we must evaluate the relation for User 4 to User 3 first, and then reverse the relation. That is,

$$a_{43} = \lfloor \frac{K_4}{t_4^2} \rfloor \bmod t_4 = \lfloor \frac{58}{4^2} \rfloor \bmod 4 = 3 \bmod 4 = 3,$$

and reverse(3) = 5, which is correct.

Now, a new user is inserted to the system, say User 7. Figure 5(a) shows this condition. Figure 5(b) shows the corresponding matrix representation of Figure 5(a).



(a) The user hierarchy structure after the new User 7.

user j

user i	1	2	3	4	5	6	7
1	0	1	0	4	2	2	0
2	1	0	2	4	0	0	3
3	0	4	0	5	0	0	2
4	2	2	3	0	3	3	6
5	4	0	0	5	0	1	0
6	4	0	0	5	1	0	0
7	0	5	4	7	0	0	0

0: No relation
1: Brother of
2: Father of
3: Grandfather of
4: Son of
5: Grandson of
6: Grand-grandfather of
7: Grand-grandson of

(b) The matrix representation.

Figure 5.

In our scheme, whenever a new user is inserted into the user hierarchy system, the corresponding key pair will be determined immediately without changing any previously defined key pairs. That is, by Theorem 3.1, $t_7 = 8$, and a new key K_7 is computed as $K_7 = 0 * 8^0 + 5 * 8^1 + 4 * 8^2 + 7 * 8^3 + 0 * 8^4 + 0 * 8^5 = 3880$. Therefore, the new key pair is $(K_7, t_7) = (3880, 8)$.

4. THE PERFORMANCE COMPARISONS

The method proposed by Wu *et al.* [13] adopts the key-key pair mechanism to determine the relationships in a user hierarchy structure; however, there still exist some drawbacks by employing their mechanism. First, their method needs to record a set of key vectors. Thus, when the number of users becomes larger and larger, the key vectors will also gradually be increased. In other words, their method needs to spend $O(n^2)$ space to store all the key vectors, where n is the total number of users. Besides, their mechanism must solve a large set of linear equations for obtaining the corresponding key vectors. If the determinant of the obtained matrix of the set of key vectors is zero, they must reassign the variables in the matrix until the resulted matrix is nonsingular.

Later, Chang [4] proposed another user hierarchy mechanism based upon the Chinese remainder theorem for n users, in which all keys K_i's are upper bounded by $q_1 \cdot q_2 \cdot \ldots \cdot q_n$, where q_i, $1 \leq i \leq n$, is a coprime integer associated to User i and $q_i > \max\{h_{ij}\}_{1 \leq i,j \leq n}$. Without loss of generality, let $q_1 < q_2 < \cdots < q_n$. The main disadvantage of Chang's method is that keys grow very rapidly when the total number of users becomes large. As to our scheme, each K_i is computed as

$$\sum_{m=1}^{i-1} a_{im}\, t_i^{m-1} \leq \sum_{m=1}^{i-1} (t_i - 1)\, t_i^{m-1} = t_i^{i-1} - 1 \leq q_1^{n-1} - 1.$$

Comparing to Chang's scheme, the upper bound of our keys is smaller than that of Chang's scheme. Furthermore, our method only need to perform a simple formula to compute the key pairs for all users. However, Chang's method uses many multiplication operations of large prime numbers, it is time consuming and impractical.

5. CONCLUDING REMARKS

A new user hierarchy mechanism based on key pairs is proposed in this paper. For each user, an integer key pair is given. In our scheme, we can easily reveal the relationship between two users by performing a simple formula. Besides, when the number of users becomes large, the implement of our scheme is easier than those of Wu *et al.*'s and Chang's schemes. Furthermore, in our scheme, a new user can be easily added to the system without reconstructing all terms of the existing key values and the generation of a new key value is simple and quick, while in the user hierarchy schemes proposed by Wu *et al.* and Chang, they cannot solve this inserting problem. However, all proposed approaches could not directly handle other relationships, such as cousin, niece/nephew, uncle/aunt except use the "common" brother to access, but this is still very hard.

REFERENCES

1. S.G. Akl and P.D. Taylor, Cryptographic solution to a problem of access control in a hierarchy, *ACM Transactions on Computer System* **1** (3), 239–247 (August 1983).
2. C.C. Chang, On the design of a key-lock-pair mechanism in information protection systems, *BIT* **26** (4), 410–417 (1986).
3. C.C. Chang, An information protection system scheme based upon number theory, *The Computer Journal* **30** (3), 249–253 (1987).
4. C.C. Chang, On the Implementation of user hierarchy structure in information system, *Proceedings of International Conference on Computer Software and Applications, IEEE*, Tokyo, Japan, pp. 412–415, (October 1987).
5. R.W. Conway, W.L. Maxwell and H.L. Morgan, On the implementation of security measures in information systems, *Communication of ACM* **15** (4), 211–220 (1972).
6. G.S. Graham and P.L. Denning, Protection-principles and practices, *Proceedings of the AFIPS Spring Joint Computer Conference*, Vol. 40, pp. 417–429, Montvale, N.J., (1972).

7. E. Gudes, The design of a cryptography-based secure file system, *IEEE Transactions on Software Engineering* **SE-6** (5), 411–419 (September 1980).
8. J.K. Jan, A single-key access control scheme in information protection systems, *Information Sciences* **51** (1), 1–11 (1990).
9. J.K. Jan, C.C. Chang and S.J. Wang, A dynamic key-lock-pair access control scheme, *Computers & Security* **10** (2), 129–139 (1991).
10. C.H. Lin, R.C.T. Lee, and C.C. Chang, A dynamic access control mechanism in information protection systems, *Journal of Information Science and Engineering* **6** (1), 25–35 (1990).
11. S.T. Mackinon, P.D. Taylor, H. Meijer, and S.G. Akl, An optimal algorithm for assigning cryptographic keys to control access in a hierarchy, *IEEE Transactions on Computers* **C-34** (9), 797–802 (September 1985).
12. J.H. Saltzer and M.D. Schroeder, The protection of information in computer systems, *Proceedings of the IEEE*, Vol. 63, pp. 1278–1308, (September 1975).
13. M.L. Wu and T.Y. Hwang, Access control with single-key-lock, *IEEE Transactions On Software Engineering* **SE-10** (2), 185–191 (March 1984).