AN ACCESS CONTROL METHOD WITH SUBJECT-OBJECT KEY AND TIME STAMP

Md. Rafiqul Islam
Computer Science and Engineering Discipline
Khulna University, Khulna –9208, Bangladesh
email: cseku@khulna.bangla.net

ABSTRACT

An access control method with subject-object key and time stamp is proposed. In this method each subject and object is assigned one key respectively. The key of a subject or an object can be used to reveal the access rights to the objects depending on the value of time stamp. The method achieves full dynamism. The operations of changing an access right, inserting a subject or an object and deleting a subject or an object need only modification of one key.

Keywords: Access Right, Subject-Object Key, Time Stamp

1.0 INTRODUCTION

Information protection is a very important issue in a computer system due to the increasing complexity of various sorts of information, the large number of users, and the widely used computer networks. The access control system acts as a framework for describing the protection mechanism. The initial system was introduced by Graham and Denning [1]. In the system, the state of an information protection mechanism is defined by a triplet \((S, O, A)\), where \(S\) is the set of subjects which are active entities of the system, \(O\) is the set of protected objects and \(A\) is an access control matrix, in which each column consists of subjects representing users or programs, and each row consists of objects representing files or records. An entry \(a_{ij}\) for \(A[S, O]\) denotes the right of subject \(S_i\) to access object \(O_j\). The access right defines the kind of authorized access to the object. An example of an access control matrix is given in Table 1. Here all access rights are expressed by numerals. Linear hierarchy of access rights may be applied here. That means, the right to read implies the right to execute, the right to write implies the rights to read and execute, and so on. In the access matrix shown below, the subject \(S_1\) can delete the object \(O_1\) and execute the object \(O_2\) and \(S_3\) can read the object \(O_3\).

<table>
<thead>
<tr>
<th>Subject</th>
<th>Object</th>
<th>(O_1)</th>
<th>(O_2)</th>
<th>(O_3)</th>
<th>(O_4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S_1)</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>(S_2)</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>(S_3)</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>(S_4)</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

0: No access, 1: Execute, 2: Read, 3: Write, 4: Delete

Based on the concept of access control matrix in 1987 Jan proposed a single key access control scheme [3]. The scheme is simple and easy to implement. In 1991 Jan et. al. proposed a two-key-lock access control system to achieve full dynamism [5]. That means, when a subject or an object is added to the system, construction of one key-lock is sufficient. On the other hand when a subject or object is deleted from the system, deletion of the key-lock is enough for necessary update. Hwang et al. proposed another two-key-lock systems using time stamp concept [6, 7]. Jan et al.’s scheme suffers problem to maintain full dynamism that is shown in Hwang’s paper [7]. Here by exploiting the single key access control scheme of Jan’s [3] and Hwang’s two-key-lock system and time stamp concept [6, 7], an access control method with subject-object key and time stamp is proposed. The proposed method is simple and achieves full dynamism in the sense that performing one inserting, deleting or updating operation need only modification of one key (subject key or object key). So, the time complexity of each operation is very
dynamic. The proposed scheme uses only $O(m + n)$ memory space for $m$ subjects and $n$ objects. To understand the two-key-lock system and time stamp concept, Hwang’s method [7] is reviewed in the next section.

2.0 ACCESS CONTROL SCHEME BASED ON CHINESE REMAINDER THEOREM AND TIME STAMP CONCEPT

In this section Hwang’s two-key-lock scheme based on Chinese remainder theorem is briefly reviewed [7]. The scheme consists of two tables, subject key-lock table and one object key-lock table. The subject (object) key-lock table has three columns: key value column, lock value column and column for time stamp. When a subject is added, the system assigns a distinct time stamp value (number) to the subject and select a prime number as lock of the subject. The key value of the subject $S_i$ ($i$th subject) is computed as follows:

$$K_i = \sum_{j=1}^{n} a_{ij} Q_j b_j \mod L'$$  \hspace{1cm} (2.1)

Where $L' = \prod_{j=1}^{n} L'_j$ (Product of all object lock values), $Q_j = L'/L'_j$, and $n$ is the total number of object in the scheme.

That means, there are $n$ such $Q_j$’s. In the system $b_j$ satisfies $Q_j b_j \mod L' = 1$. Therefore, $b_j = \text{inv} (Q_j, L')$. To calculate $\text{inv} (Q_j, L')$ the extended Euclid’s algorithm is required [1, 2]. Access right is computed as follows:

$$a_q = K_i \mod L'$$  \hspace{1cm} (2.2)

When an object is added to the system its key value is computed as follows:

$$K'_j = \sum_{i=1}^{m} a_{ij} Q_i b_i \mod L$$  \hspace{1cm} (2.3)

Where $L = \prod_{i=1}^{m} L_i$ (Product of all subject lock values), $Q_i = L'/L_i$, and $m$ is the total number of subject in the scheme. From object key value access right is computed as

$$a_q = K'_j \mod L_i$$  \hspace{1cm} (2.4)

Using the above equations the subject and object key-lock tables are constructed. To construct the key-lock tables for subjects and objects, we consider the access matrix of Table 1. Suppose that the subjects and objects are added to the scheme in the sequence $S_1, O_2, O_2, S_2, O_3, S_3, O_4$. Let $T_S$ is the time stamp of the subject $S_i$ and $T_O$ is the time stamp of the object $O_i$. In Table 2 $K_i$ is the key and $L_i$ is the lock of the subject $S_i$ respectively. In the Table 3 $K'_j$ is the key and $L'_j$ is the lock of the object $O_j$ respectively. The lock values are relatively pairwise prime numbers.

Table 2: The subject key-lock table

<table>
<thead>
<tr>
<th>Subject</th>
<th>$K_i$</th>
<th>$L_i$</th>
<th>$T_S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>Null</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>$S_2$</td>
<td>7</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>$S_3$</td>
<td>1</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>$S_4$</td>
<td>7</td>
<td>11</td>
<td>6</td>
</tr>
</tbody>
</table>
### Table 3: The Object key-lock table

<table>
<thead>
<tr>
<th>Object</th>
<th>$K_j$</th>
<th>$L_j$</th>
<th>$TO_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O_1$</td>
<td>4</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>$O_2$</td>
<td>4</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>$O_3$</td>
<td>135</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>$O_4$</td>
<td>246</td>
<td>11</td>
<td>7</td>
</tr>
</tbody>
</table>

#### 2.1 Verification of Access Right

To verify the access right of subject $S_i$ to the file $O_j$, the time stamp numbers $TS_i$ and $TO_j$ of subject and object is compared. If the time stamp number of the subject is less than that of the object, i.e., subject $S_i$ is added to the system before the object $O_j$, the system uses the lock of value of the subject and the key value of the object to verify the access right of the subject to the object. On the other hand, if the time stamp value of the subject is greater than that of the object, i.e., the subject $S_i$ is added to the system after the object $O_j$, the system uses the key value of the subject and the lock value of the object to verify the access right of the subject to the object.

**Example 2.1: Verification of Access Right**

Suppose that $S_3$ wants to execute the object $O_4$, the system fetches the time stamp $TS_3$ and $TO_4$ from the subject and object key-lock tables. Here $TS_3 = 4$ and $TO_4 = 7$, that means $TS_3 < TO_4$, so $a_{34} = K_4' \mod L_3 = 246 \mod 7 = 1$.

Since $a_{34} = 1$ is equal to the requested access right 1 (execute), the access request is permitted. On the other hand if $S_4$ wishes to write something in the object $O_1$, the system compares $TS_4$ and $TO_1$ and finds $TS_4 > TO_1$ (since $TS_4 = 6$ and $TO_1 = 1$). So, $a_{41} = K_4 \mod L_1 = 7 \mod 5 = 2$.

Since $a_{41} = 2$ (read) is less than the requested access right 3 (write), the access request is denied.

In this scheme the key construction process is time consuming due to $Q_i$, $b_i$, $Q_j$ and $b_j$. The interested readers may see the simulation results of such computations in the paper [8]. Using the concept of time stamping and the single key method a simple and dynamic subject-object key access control method is proposed here. The key construction process of the method is simple and verification of access right is easy. On the other hand the system achieves full dynamism. The proposed scheme is introduced in the next section.

### 3.0 A METHOD WITH SUBJECT-OBJECT KEY AND TIME STAMP

In this section the proposed method is described with respect to the key construction process, verification of access right and dynamic access control, such as changing access right, adding a subject or an object and deleting a subject or an object from the system.

#### 3.1 The Key Construction Process

Suppose that there are $m$ subjects and $n$ objects in the system currently. Here $a_{\text{max}}$ is the maximum value of access rights of system ($a_{\text{max}} = 4$ according to Table 1). The system consists of two tables, one subject key table and one object key table. The subject key table contains two columns: key value column and time stamp column. Similarly, the object key table has two columns: key value column (key of the object) and column for time stamp number. The key of a subject is computed from access rights of the subject to objects and the key of an object is computed from the access rights of subjects to the object (the object for which the key value is computed).

The key of the subject $S_i$ is computed as follows:

$$K_i' = \sum_{j=1}^{n} a_{ij} \cdot R^{j-1}$$  \hspace{1cm} (3.1)

Where $R = a_{\text{max}} + 1$, $n$ is the number of objects in the current system.
The key of the object $O_j$ is computed as follows:

$$K_j = \sum_{i=1}^{n} a_{ij} R^{i-1}$$  \hspace{1cm} (3.2)

Where $m$ is the number of subjects in the current system.

Example 3.1: Construction of subject and object key tables
Let us consider the access control matrix of Table 4. Let $S_1, S_2, S_3$ and $O_1, O_2, O_3, O_4$ be the three subject and the four objects which are added to the system in the sequence $S_1, O_1, O_2, O_3, S_3, O_4$. Considering the corresponding access rights of the Table 4 we compute the keys of the subjects (objects) and their time stamps as follows:

$TS_1 = 0; \quad K_1 = 0$

$TO_1 = 1; \quad K'_1 = 1$

$TO_2 = 2; \quad K'_2 = 2$

$TS_2 = 3; \quad K_2 = 2$

$TO_3 = 4; \quad K'_3 = 3 \times 5 = 15$

$TS_3 = 5; \quad K_3 = 4 \times 5 = 20$

$TO_4 = 6; \quad K'_4 = 4 + 2 \times 5^2 = 54$

Table 4: Access Control Matrix

<table>
<thead>
<tr>
<th>Object</th>
<th>Subject</th>
<th>$O_1$</th>
<th>$O_2$</th>
<th>$O_3$</th>
<th>$O_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>$S_2$</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$S_3$</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

By the above method we maintain two key tables for subjects and objects as presented in Table 5 and Table 6.

Table 5: The subject key table

<table>
<thead>
<tr>
<th>Subject</th>
<th>$K'_j$</th>
<th>$TS_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$S_2$</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>$S_3$</td>
<td>20</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 6: The object key table

<table>
<thead>
<tr>
<th>Object</th>
<th>$K'_j$</th>
<th>$TO_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O_1$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$O_2$</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$O_3$</td>
<td>15</td>
<td>4</td>
</tr>
<tr>
<td>$O_4$</td>
<td>54</td>
<td>6</td>
</tr>
</tbody>
</table>

3.2 Verification of Access Right

The access right of the subject $S_i$ to the object $O_j$ is computed as below:

$$a_{ij} = \begin{cases} 
\left[ \frac{K_i}{R^{j-1}} \right] \mod R & \text{if } TS_i > TO_j \\
\left[ \frac{K'_j}{R^{i-1}} \right] \mod R & \text{if } TS_i < TO_j \end{cases}$$  \hspace{1cm} (3.3)

Where $TS_i$ is the time stamp number of subject $S_i$ and $TO_j$ is the time stamp number of object $O_j$. 

80
Example 3.2: Access Right Verification

Suppose that the Subject $S_2$ wants to write in the object $O_3$. Now we have to verify whether the subject $S_2$ is permitted to write in the object $O_3$ or not. Here $T_{S_2} = 3$ and $T_{O_3} = 4$, i.e., $T_{S_2} < T_{O_3}$. Using equation (3.3) we get

$$a_{23} = \left\lfloor \frac{K_j}{R^{j-1}} \right\rfloor \mod R = \left\lfloor \frac{15}{5} \right\rfloor \mod 5 = 3 \mod 5 = 3$$

Since, the system found $a_{23} = 3$ (write) which is equal to the requested access right 3 (write), the request of the subject is permitted.

If the subject $S_3$ sends a request to read the object $O_1$ we have to get $a_{31} = 1$ (read) from the system. Here $T_{S_3} = 5$ and $T_{O_1} = 1$, i.e., $T_{S_3} > T_{O_1}$. Now from the equation (3.3) we get

$$a_{31} = \left\lfloor \frac{K_j}{R} \right\rfloor \mod R = 5 \mod 5 = 0$$

$a_{31} = 0 \neq 1$, so the request will be denied by the system.

### 3.3 Changing Access Right

When the access right of subject $S_i$ to object $O_j$ is changed from $a_{ij}$ into $a'_{ij}$, we first compare the time stamp values $T_{S_i}$ and $T_{O_j}$ of the subject and the object. Then we compute the new key value $K_i$ or $K_j$ using the old key value $K_i$ or $K_j$ according to algorithm given below.

Algorithm 3.1: Changing Access Right

1. Input $a_{ij}$ and $a'_{ij}$
2. If $T_{S_i} > T_{O_j}$ then
   
   $$K_i = K_i + (a'_{ij} - a_{ij}) \cdot R^{i-1}$$
   
   Else
   
   $$K_j' = K_j' + (a_{ij} - a'_{ij}) \cdot R^{j-1}$$
   
   Output new key value $K_i$ or $K_j'$.

Example 3.3: Changing Access Right

Suppose the access right $a_{21} = 2$ (see Table 4) is changed into $a'_{21} = 3$. Here $T_{S_2} = 3$ and $T_{O_1} = 1$, i.e., $T_{S_2} > T_{O_1}$. So, we have to update $K_2$. The old value of $K_2 = 2$. According to the algorithm 3.1, we get the new value of $K_2 = 2 + 1 = 3$.

### 3.4 Inserting a Subject or an Object

When a subject is inserted into the system, we assign a time stamp number as the time stamp of the subject. Then the key value of the subject is computed by equation (3.1). To insert a new object, the system assigns a time stamp numbers to the object and the key value of the object is computed by equation (3.2).

### 3.5 Deleting a Subject or an Object

The deleting process is very easy. When a subject or an object is being deleted from the system, the key value and the time stamp of the subject or object is deleted from the subject (object) key table.

### 4.0 DISCUSSION

We assume that the system has $m$ subjects and $n$ objects. By ignoring the overflow problem of key values, the space complexity of the proposed method is $O(m+n)$ such as each subject or object possesses a key. For the time complexity of the method, we assume that each arithmetic operation needs $O(1)$ time only. Equation (3.1), to construct a key of a subject, need to access all access rights of the objects. Thus its time complexity is $O(n)$. Similarly, time complexity to construct a key of an object is $O(m)$. By equation (3.3) to check access right of a user to an object need $O(1)$ time. To delete a subject or an object from the system, only its corresponding entry is
removed from the key tables, the time complexity is also $O(1)$. Here it may consider the number of modified key values for each operation. It is easy to see that the proposed method updates one key for the operation of changing access rights of a subject to an object and inserting a subject or an object to the system. On the other hand deletion process is very simple. Here the following remarks can be highlighted:

1. To reveal the access right of a subject to an object a simple operation on one key of a subject or an object is enough.
2. To change the access right of a subject to an object, it modifies only the key of the subject or object.
3. To insert a new subject/object into the system, the proposed method only construct (assigns) a key to the subject/object without modification of the other keys.
4. To delete a subject or an object from the system, it simply removes the corresponding entry of the subject or object from the subject/object key table.

One issue highlighted is that one integer may not be enough for storing one key value. For instance, if we consider a 64-bit computer, the largest integer allowed by such a computer is $2^{64}$. Since each key value is a sum of terms with power of $R$ ($R = 5$ as shown), there may be an overflow to hold one key value by one integer. In such a case the system requires special data structure such as array or record for holding one key value.

5.0 CONCLUSION

In this paper a very simple and efficient subject-object key access control scheme using time stamp is proposed. For the proposed method formulas for constructing key and verification of access right are devised, here an algorithm for updating access right is also devised. The proposed scheme achieves full dynamism, that means, changing access right, insertion and deletion of subject (object) can be implemented by performing operations on only one key. The required space for the scheme is not large.

REFERENCES


**BIOGRAPHY**

**Md. Rafiquel Islam** obtained Master of Science (M. S) in Engineering (Computers) from Azeraijan Polytechnic Institute in 1987 and Ph.D. in Computer Science from Universiti Teknologi Malaysia (UTM) in 1999. Currently he is an Associate Professor and the Head of Computer Science and Engineering Discipline of Khulna University at Khulna of Bangladesh. His research areas include Design and Analysis of Algorithms, Information Security and External Sorting. He has got a number of papers related to these areas published in national and international journals.