CHAPTER 5

SPATIO-TEMPORAL ROLE BASED ACCESS CONTROL TECHNIQUES

5.1 ACCESS CONTROL

Access Control (AC) is one of the major techniques used for protecting information stored in files and databases. It is an important component in security because it has a direct impact on the integrity and confidentiality of an information system. Moreover, access control useful in the decision making process since it is useful for authorizing users who want to access system resources. Through this mechanism, it is possible to limit access to resources based on user’s past and current behaviour status based on their membership in various predefined groups. Therefore, access control can be referred to as an authentication or identity verification process. In a classical information framework, an access control policy maps each user, resource and action to a decision. This policy is then used at all later times whenever a particular user wants to access the resource in all his future attempts.

In the past, different access control techniques have been proposed and implemented by database researchers. Among them, Role Based Access Control (RBAC) (Ravi Sandhu et al 1996) has been the subject of interest for many years and is widely accepted as an alternative to traditional discretionary and mandatory access control models. Bertino et al (2001) proposed an extension of RBAC model called Temporal Role Based Access
Control Model (TRBAC). The main features of this model are, it supports periodic enabling/disabling of roles, individual exceptions, and the possibility of specifying temporal dependencies among such actions, expressed by means of role triggers. Though many such models are available in the literature, most models do not take into consideration the environmental factors before making access decisions. Therefore, such models are not being suitable for many applications which need time and location. To overcome the deficiency in the existing models Ray and Manachai Toachoodee (2007) proposed a Spatio-Temporal Role Based Access Control (STRBAC) Model. Their models identify the entities and relations in RBAC and analyze the dependencies based on location and time. Moreover, the emergence of distributed environment in web databases poses new demands on access control mechanism, because of decisions to grant access may depend on contextual information such as the location of the user and the time which access request are made. However, none of models present in the literature are exactly suitable for web databases because of the dynamic and distributed nature of data used in web database.

In this thesis, a new Intelligent Spatio-Temporal Role Based Access Control model (ISTRBAC) has been proposed which uses intelligent agents for rule management which are used for setting privileges. Moreover, intelligent rules are used also for enforcing spatio-temporal constraints which are necessary for web databases since they have to validate and maintain integrity and security in heterogeneous and multi databases.

In this thesis, an Intelligent Spatio-Temporal Access Control Model (ISTRBAC) has been proposed by adding identity constraints and spatio-temporal constraints in access control policies. This system provides separate agents such as a spatial information agent, temporal information agent and a rule management agent to check appropriate constraints. Moreover, this thesis proposes further more new agents that are capable of providing rule matching and rule firing so that the accuracy level in providing decisions on access control is increased to an optimal level of security. In order to increase the access
to resources on the web, this thesis proposes an additional model called spatio-temporal Action Status Access Control model. The key feature of this model is that it uses intelligent agents to make decision and to take action which the requests to access resources are obtained. In this scenario, the intelligent agent’s ascribed status, action status and spatio-temporal constraints are used for decision making.

5.2 AN INTELLIGENT AGENT BASED FRAMEWORK FOR SECURE WEB DATABASES

In this research work, a new framework for implementing the proposed Intelligent Agent Based Spatio-Temporal Role Based Access Control model (ISTRBAC) that support spatio-temporal constraints on the enabling / disabling of roles has been developed.

Figure 5.1 shows the overall architecture of the intelligent agent based spatio-temporal RBAC model proposed in this research work. The functionality of the various components of the model is described as follows:

![Figure 5.1 Overall Architecture of the Intelligent Agent Based Access Control Subsystem](image-url)

Figure 5.1 Overall Architecture of the Intelligent Agent Based Access Control Subsystem
It consists of various modules such as user interface, access control manager, spatio-temporal information manager, Rule Manager, Role assignment Manager, Privacy Manager, spatio-temporal constraints, knowledge base, Role Details, Permission Details and the web databases.

In this architectural framework, various intelligent agents have been used to validate the privileges and to perform effective decision making.

5.2.1 User Interface

In this framework user interaction is performed through the user interface. It accepts user queries and validates them using a validation agent. Normal queries are identified by the user interface and are sent to the access control manager for deciding on the access privileges. On the other hand, malicious queries are filtered and dropped by the user interface.

5.2.2 Database Manager

Database Manager receives the query from the user interface and passes it to the other modules such as a rule manager, spatio-temporal information manager, user-role assignment manager and privacy manager to check spatio temporal constraints, identity constraints and status level of the user before accessing web database. If all these constraints are satisfied then the request is allowed to access the database and the query result is passed back to the user interface.

5.2.3 Rule Manager

The responsibility of rule manager is to provide various database services and to support spatio-temporal information manager for checking spatio temporal constraints. The knowledge base used by the rule manager has
the collection of active rules and event details, which automatically fire actions in response to events under the conditions defined.

5.2.4 Knowledge Base

Using the knowledge gained from the experts after validation, the knowledge is coded into the knowledge base in the form of facts and rules. Database systems provide indexing and hashing techniques to organize the data effectively in the physical database. On the other hand in this system, the entire set of rules and facts are stored in knowledge base using a frame-based Knowledge representation technique for effective creation and maintenance of the Knowledge Base. The major advantage of using this knowledge representation helps perform effective retrieval in order to reduce the cost of query processing. This Knowledge Base uses a slot-filter format to store facts and rules using two fields respectively for storing the Condition and action parts.

5.2.5 Spatio-Temporal Information Manager

It is mainly responsible for managing spatio-temporal constraints and to provide information about user’s location from where the user is trying to access the web databases and the time interval during which they are trying for data access.

5.2.6 Spatio-Temporal Constraints Base

It is designed to represent spatial and temporal constraints effectively. It provides rules to check whether it is necessary to use the temporal index structure namely time-split tree to organize the data in an optimal way in the database. Moreover, it checks whether spatial-index using R-tree has to be used for storing the spatial data. It also uses Allens Interval
algebra (Allen 1983) to perform temporal reasoning using valid time and location of the users.

5.2.7 User Role Assignment Manager

This manager consults rule manager and checks the time interval and location of the user and assigns suitable roles depending on the application requirements. Moreover, it uses roles from role base to decide on user role assignment.

5.2.8 Role Base

The role base consists of a collection of roles designed by the experts and system designers based on the requirements. These roles are coded into the role base in the form of name of the job functions and are maintained for effective usage by the system.

5.2.9 Privacy Manager

The privacy manager takes care of privacy related issues by way of assigning permissions to the users based on spatio temporal constraints, purpose, obligation and conditions.

5.2.10 Permission Details Base

The permission details database is used to store details of the permissions and related details such as obligation, purpose and condition. These details are used by the privacy manager to assign permission to the new roles after checking appropriate rules. Therefore it, communicates and coordinates with the rule manager for effective decision making on permission assignment. Finally this database helps the system for providing permission assignment details effectively in order to solve the privacy related issues.
5.2.11 Web Database

It is a storage structure designed to represent the web pages effectively. Moreover, it supports site operations and to manage collection of web data by using suitable storage structures. It helps the web users to collect large amount of data from various locations. Finally it supports the database manager for performing data manipulation using dynamic functions and for storing domain dependent information. Moreover, it uses two types of indexing methods namely, full indexing and human indexing to get search results faster.

5.3 THEORETICAL MODELING

The theoretical modeling of the spatio-temporal access control model proposed in this thesis explained in this section. For this purpose, a few definitions are introduced here for explaining the proposed model. This model has been proposed by modifying the existing model (Xiutao Cui et al 2007) with location and interval constrains.

**Definition 1** Spatio-Temporal Database (D): It is the collection D of spatio-temporal data stored in the used in an interaction, spatio-temporal database, web database and user interface etc. Let D is the set of spatio-temporal data.

**Definition 2** Operation (OP): It is the data definition, data manipulation and data control operations carried out on the spatio-temporal data. Let OP be the set of operations, each element in OP denotes a kind of execution on data. Let DM be the subset of OP, i.e. DM ∈ OP. Here DM indicates Data Manipulation operation.

**Definition 3** Privilege (P): It is specification of the data control operation set on the spatio-temporal data. Let P be the set of all access privileges provided on a database object to a user during time interval \((t_1, t_2)\) and from location \((l_n, l_a)\). If p \((d, DM, (t_1, t_2, l_n, l_a)) \in P\) where d ∈ D, DM ∈ OP,
Definition 4 Distributed Collaboration Role (DCR): It is a group of users with the same duty in distributed collaboration, for example in group visit activity of web sites (database), there are 4 collaboration roles: member supervisor, service provider and naive user. Let DCR be set of distributed collaboration role. Then each user is provided with a trusted key $k_i$ for performing communication.

Definition 5 Distributed Interaction Role (DIR): It is a group of distributed users having similar role with same privilege set in an interaction. Let DIR be the set of distributed interaction roles provided an application program for interaction between the distributed roles.

Definition 6 Identity Constraint (ID): It is used to specify which distributed collaboration role can be assigned to a given interaction role depending upon the application and the data needs. Let IC be the set of all identity constraints for a distributed interaction role. These constraints are to be checked and satisfied before any data access.

Definition 7 Spatio-Temporal Constraint (Cst): It specifies under what spatio-temporal relationship the user can be assigned to a given distributed interaction role. Let $\text{Cst} = (k_i, \text{sr}, (\Delta t_1, \Delta t_2))$ be the spatio-temporal constraint, which sr denotes spatial relationship, $(\Delta t_1, \Delta t_2) \geq 0$ denotes interval time extent and $k_i$ denotes the key value. The spatio-temporal constraint can be described: the user must have been kept the spatial relationship sr for the time extent $\Delta t$ at least.

Definition 8 Distributed Interaction Role Mapping Set (DIRM): Let M be the set of mapping relationships between constraints (IC, Cst) and
distributed interaction role DIR. Therefore, M can be defined as
\{f(m): (IC \times Cst) \rightarrow \text{DIR}, m \in M\} which describes who and under what
identity constraints and spatio-temporal relationship can be assigned a given
interaction role in an interaction.

**Definition 9** (Spatio-temporal Satisfaction): It means that users \(k_i\) at location
\((l_n, l_a)\) and time interval \((t_1, t_2)\) satisfies the spatio-temporal constraint of a
distributed interaction role, denoted: \(f(k_i, l_n, l_a, t_1, t_2) = m_i\) for \(1 \leq i \leq n\), where
\(n\) is the number of tuples present in the database.

**Definition 10** (Distributed User): It is the person who attends the interaction
activity through a distributed interaction role when user satisfies the identity
constraint and spatio-temporal constraints. The \(i^{th}\) user is identified using the
primary key \(k_i\).

### 5.3.1 Rule in User Role Assignment

The user is assigned to role \(r\) during time interval \((t_1, t_2)\) and if the
user is within the specified area of location \((l_n, l_a)\). To this hold, the location
of the user when the role was assigned must be in any one of the location
within the specified area specified by a circle with centre \((l_n, l_a)\) and radius \(r_1\)
where the role allocation can take place. Moreover, the time of role
assignment must be between \(t_1\) and \(t_2\) when role allocation can take place.

**Rule 1**

Given a user \(u\) with \(k_i \in D, (l_n, l_a) \in L, r \in R\) and
\((t_1, t_2) \in T\) (Temporal Constraints)

If UserLocation \((k_i, \ell) = (l_n, l_a)\)
and start time and end time are \((t_1, t_2)\)
Then the user \(k_i\) is assigned to role \(r\).
5.3.2 Rule in Role Activation

The user $u$ with $k_i$ is activated role $r$ during the interval $(t_1, t_2)$ at location $(l_n, l_a)$ if the predicate $\text{UserRoleAlloc} (k_i, (l_n, l_a), r, (t_1, t_2))$ is true. This predicate implies that the location of the user during the role activation must be a subset of the allowable locations for the activated role and time interval when the role remains activated must be belong to the duration when the role can be activated and the role and role can be activated only if it is assigned.

Rule 2

Given a user $u$ with $k_i \in D$, $(l_n, l_a) \in L$, $r \in R$ and $(t_1, t_2) \in T$

If user key = $k_i$ and

$(l_n, l_a) \subseteq \text{RoleEnableLoc} (r) \in L$ AND

$(t_1, t_2) \subseteq \text{RoleEnableDur} (r) \in DT$ AND

UserRole Assignment ($k_i$, $l_n$, $l_a$, $r$, $t_1$, $t_2$)

THEN UserRoleActivated

5.3.3 Rule in Permission Assignment

Permissions are associated with roles, objects and operation. Three additional entities with permission to deal with spatial and temporal constraints have been associated here namely user location, object location and time interval $(t_1, t_2)$. Now a rule for permission assignment is defined as follows.
Rule 3

Given a user u with $k_i \in D$, $(f_n, f_a) \in L$, $r \in R$ and $(t_1, t_2) \in T$, role enabled
If
$$(f_n, f_a) \subseteq (\text{PermRoleLoc}(p, r) \ \text{AND} \ \text{RoleEnableLoc}(r)) \ \text{AND}$$
$$(t_1, t_2) \subseteq (\text{permDur}(T) \ \text{AND} \ \text{RoleEnableDur}(t_1, t_2) \in DT)$$
THEN Permissions Assigned (p)

5.3.4 Rules in Conflict Role Assignment

In this section, we propose some rules that Ex-STRABC model proposed in this work must obey by the analysis of different conflicts when we assign the privilege to distributed interaction role, and assign the interaction role to user.

Definition 11 Operation Conflict): For all $k_1, k_2 \in D$, $DM_1, DM_2 \in OP$, If any user u with key $k_1$ cannot execute op on $k_1$ and execute op$_2$ on $k_2$ during the same time between $t_1$ and $t_2$, we call that op$_1$ on $k_1$ is conflict with op$_2$ on $k_2$, denoted: $p_1 \otimes p_2$.

The compatibility matrix of operations is as follows:

Table 5.1 Compatibility Matrix for Operations

<table>
<thead>
<tr>
<th>OP</th>
<th>R</th>
<th>W</th>
<th>WI</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>W</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>WI</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>

where R-Read, W-Write and WI- Write Intent
Rule 4  Given a privilege \( p = (k_i, DM, t_1, t_2, f_n, f_a) \), if for all \( op_1 \in DM, op_2 \in DM, (t_1, t_2) \in T \) and \( (f_n, f_a) \in L \) then it doesn’t exist conflict between \( op_1 \) and \( op_2 \) on data set \( k_i \).

**Definition 12** Identity Constraint Conflict: For all \( dcr \in DCR, dcr_1, dcr_2 \in DCR, \exists (m_1, m_2 \in M) (m_1.dr = dcr_1, m_2.dr = dcr_2) \land (dcr \notin (m_1.IC \cup m_2.IC)) \), we call the identity constraint of the two interaction role: \( ir_1, ir_2 \) is conflict on distributed collaboration role \( dcr \). Denoted as \( m_1.IC \otimes m_2.IC \).

Rule 5  Given a user \( u \) with key \( k_i \), \( \forall d ir_1, d ir_2 \in k_i.CI \), where \( CI \subseteq DIR, \exists (m_1, m_2 \in M) \land (m_1.r = dir_1, m_2.r = dir_2) \land (\neg (m_1.IC \otimes k_i.dcr m_2.IC)) \), Denoted: IC Compatible \( (k_i, \{dir_1, dir_2\}) \), that is to say any two interaction roles assigned to a user do not have identity constraint conflict with each other.

**Definition 13** Spatio-Temporal Constraint Conflict: If two spatio-temporal constraints cannot co-exist at the same time, we call that \( C_{st1} \) is conflict with \( C_{st2} \) denoted as \( C_{st1} \odot C_{st2} \).

Rule 6  Given a user \( u \) with key \( k_i \), \( \forall dir_1, dir_2 \in k_i.CI \exists (m_1, m_2 \in M) \land (m_1.r = dir_1, m_2.dr = dir_2) \land (\neg (m_1.Cst \otimes m_2.Cst)) \), Denoted as ST Compatible \( (k_i, \{dir_1,dir_2\}) \), that is to say any two interaction roles assigned to a user do not have spatio-temporal constraint conflict with each other.

5.4 RULES FOR ROLE VALIDITY CHECKING

Due to the changing of data location in a distributed collaboration, the data’s location might not satisfy the spatio – temporal constraints of interaction roles any more or vice versa. Therefore, with the passing of time and the change of user’s location, we need to check the validity of interaction roles dynamically, and thus grant or revoke a given interaction role of a given user.
To check the validity of interaction role dynamically, we give the following three different validity checking rules.

### 5.4.1 Identity Constraints Checking Rule

\[ \exists m \in M, (m. r = r) \land (k_i. dcr \in m.IC), \text{i.e. The collaboration role of the user } u \text{ must satisfy the identity constraint of interaction role } ir, \text{ the algorithm of identity checking rule is showed below:} \]

```java
RoleCRCheck (user with key k_i, role r)
{
    for each m in M
    if k_i.dcr = m.IC then
        if m.ir \in DIR then
            return true
        else
            return false
    }
}
```

### 5.4.2 Spatio-temporal Constraints Checking Rule

\[ \exists m \in M, (m. ir = ir) \land ((k_i, (l_n, l_a), k_i, (t_1, t_2)) \Rightarrow m.Cst), \text{i.e. The location } (l_n, l_a), \text{ of user } k_i \text{ and during time interval } (t_1, t_2) \text{ and location } (l_n, l_a), \text{ must satisfy the spatio-temporal constraint } Cst \text{ of an interaction role, the algorithm of spatio-temporal checking rule is showed below:} \]

```java
DIRoleSTCCheck (user u, role ir)
{
    for each m in M
        if m.ir = ir then
            if (k_i, (l_n, l_a), k_i, (t_1, t_2)) \Rightarrow Cst then
            return true
        else
            return false
    }
}
```
return true
else
    return false
}

5.4.3 Conflict Checking Rule

\forall i, r_{k_i} \in k_i.CI, \neg(\text{ir}_{k_i} \otimes k_i.dcr \ ir), i.e. the distributed interaction role to be assigned cannot conflict with Any of interaction role iru which user u must satisfy the identity constraint of interaction user u must satisfy the identity constraint of interaction checking rule is shown below:

IRoleConflictCheck (user \ k_i, role r)
{
    for each r_{k_i} in k_i.DIR
        if dr_{k_i} \otimes k_i.dcr \ ir
            then
                return true
        return false
}

According to these three validity checking rules, we perform periodic checking rules. First, we use the spatio-temporal checking by applying a rule to check whether the user’s location and time satisfy the spatio-temporal constraint of a distributed interaction role assigned to a user. If it is false, then we revoke the distributed interaction role from the user; secondly, we use all these checking rules to check whether the user satisfies the identity constraint, spatio-temporal constraint and conflict constraint of
other interaction roles. If that is true (i.e. no conflict) then we grant the distributed interaction role to the user.

5.4.4 Operation Validity Verification

When a user attempts to execute an operation on data, (s)he needs send a request of operation executing to the access control server. The access control server verifies the validity of request according to operation validity rule.

**Definition 14** Operation Request: Let \( OREQ = (k_i, op, d, IC) \) be an operation executing request, i.e., it denotes an operation executing request from user \( k_i \) on data \( d \). The operation validity rule is showed below:

Role Operation Validity Rule: Given an operation \( op \) of the distributed interaction role \( ir \) on data \( d \), if \( \exists (d, DM) \in ir.P, op \in DM \), then we can say that the operation \( op \) of the role \( ir \) on data \( d \) is validity, denoted: \( (d, op) \rightarrow ir \).

User Operation Validity Rule: Given an operation \( op \) of the user \( k_i \) on data \( d \), if \( \exists ir \in k_i.DIR, (d, op) \rightarrow ir \), then we can say that the operation \( op \) of the user \( u \) with key \( k_i \) on data \( d \) is valid, denoted as \( (d, op) \rightarrow u \).

**Definition 15** Operation Response: Let \( ORES = (k_i, op, d, result) \) be the response of operation executing request, i.e., it denotes the response of an operation executing request from user \( u \) with key \( k_i \) on data \( d \).

5.5 IMPLEMENTATION OF INTELLIGENT AGENT BASED SPATIO-TEMPORAL RBAC

To implement the access control policies with spatio-temporal constraints, first the system obtains the user ID and password from the users
when they try to enter the system. Now the user’s privileges are checked with rule base where the details of all users have been stored with their access privileges. The user permissions are validated with the access privileges and in addition, the spatio-temporal constraints are used for validating spatio-temporal access control policies.

The access control subsystem has been implemented using roles assigned periodically that consider spatio constraints as well. In order to specify the roles and process the access control policies, an algorithm called ISTRBAC algorithm has been proposed and implemented in this research work. The steps of this algorithm are as follows:

Step1 : Initialize Queue for data access request

Step2 : Read the user request

Step3 : Check (user_previlege, Database_previlege)

Step4 : Check for the access rights using the temporal and spatial constraints.

Step5 : If the user is a valid user and the permission are assigned already to the client after checking the spatio-temporal constraints,

    Permit (access, object, t1,t2, t_m, t_a)

    User_try_count = user_try_count + 1;

    Auth_user_count = auth_user_count + 1;

    Else

    User_try_account = user_try_count + 1;

    Deny the access,
Step6 : Check for Queue empty,

Step7 : Display (user_try_count, auth_user_count).

Based on the Implementation of the algorithm for spatio-temporal role based access control with intelligent agents, the system has been tested with various datasets for temporal constraints. Table 5.2 shows that the number of authorised users who were denied access by the ISTRBAC model to access the database during the interval \((t_1, t_2)\) and when they are trying to access from the location \((l_n, l_a)\).

From Table 5.2, it can be observed that the proposed ISTRBAC model performs better when compared with RBAC model in restricting the users.

**Table 5.2 Number of User Requests Denied Access by RBAC and ISTRBAC**

<table>
<thead>
<tr>
<th>Exp.No</th>
<th>No.of User Request Tried</th>
<th>No.of requests denied by RBAC</th>
<th>No.of requests denied by ISTRBAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp1</td>
<td>100</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Exp2</td>
<td>200</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>Exp3</td>
<td>300</td>
<td>10</td>
<td>13</td>
</tr>
<tr>
<td>Exp4</td>
<td>400</td>
<td>13</td>
<td>17</td>
</tr>
<tr>
<td>Exp5</td>
<td>500</td>
<td>14</td>
<td>22</td>
</tr>
</tbody>
</table>

Figure 5.2 shows the number of authorized users who were permitted by the intelligent agent based ISTRBAC model. From this, it is observed that the access permission of ISTRBAC is lower than RBAC model. Moreover, 5% of less users where denied access in comparison with the existing system and hence the security is enhanced.
Figure 5.2  The Number of User Requests Permitted after Checking Access Permission Using RBAC and ISTRBAC

Table 5.3 shows the number of users requests denied access by the ISTRBAC model after checking identity and spatio-temporal constraints to access the database.

**Table 5.3  The Number of User Request Denied Access by TRBAC and ISTRBAC Model**

<table>
<thead>
<tr>
<th>Exp. No</th>
<th>No. of User Request Tried</th>
<th>No. of requests denied by TRBAC</th>
<th>No. of requests denied by ISTRBAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp1</td>
<td>100</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>Exp2</td>
<td>200</td>
<td>9</td>
<td>13</td>
</tr>
<tr>
<td>Exp3</td>
<td>300</td>
<td>12</td>
<td>20</td>
</tr>
<tr>
<td>Exp4</td>
<td>400</td>
<td>17</td>
<td>27</td>
</tr>
<tr>
<td>Exp5</td>
<td>500</td>
<td>21</td>
<td>34</td>
</tr>
</tbody>
</table>

Figure 5.3 shows the number of user request permitted by traditional TRBAC Model and intelligent agent based ISTRBAC Model when the request is sent during time interval \((t_1, t_2)\) and from the location \((l_n, l_a)\).
From the implementation carried out in this model, it is observed that there is a difference of 7% users who were denied access in comparison with the TRBAC model.

![Figure 5.3 The Number of Users Permitted During Time Interval (t₁, t₂) and from the Location (fₐ, fₐ)](image)

5.6 SPATIO-TEMPORAL ACTION STATUS ACCESS CONTROL

In this work, the action status access control model proposed by Barker (2008) has been extended with intelligent agents and spatio-temporal constraints. The main aim of original action status access control model is to fulfill the requirements of access policy representation for dynamic and distributed information systems. In this model, a user is viewed as a rational entity the t can, within certain constraints, choose the actions it perform in order to increase its access to resources by changing action status. A key feature of this model is that a decision on an agent’s request to access resources is determined by considering the agent’s ascribed status and agent’s action status which is determined from a history of the deliberative actions performed by the agent. An agent’s ascribed status together with the agent’s action status gives a measure of the agent’s overall status level. The agent’s
status level is used as the basis for determining authorized actions and this is used in rendering decisions on the agent’s access request.

5.6.1 Model Description

In this work, a spatio-temporal access control model has been proposed and implemented. For this purpose, some of the key sets of constants in the alphabet that are used in the formulation of the Spatio-Temporal Action Status Action Control (STASAC) model are as follows:

- Accountable set \( O \) of object identifiers.
- Accountable set \( A \) of named actions.
- Accountable set \( SL \) of status level identifiers.
- Accountable set \( E \) of event identifiers.
- Accountable set \( U \) of agent (subject, user) identifiers.
- Accountable set \( T \) of timepoints and duration time
- Accountable set \( L \) of locations

In this spatio-temporal model’s interpretation, an object is anything about which data are collected and stored in the database. For example, individual facts within a database for a bank account in which attributes are used for describing their properties. All objects have a unique identity that is invariant and will be the primary key of the object relation. In addition some objects may have properties which are described by attribute name, data type and constraints. The set \( A \) of actions includes (i) the actions that agents may perform to change a status level assignment (e.g. actions like depositing money, joining a loyalty scheme, etc), and (ii) the actions the agent can perform as a consequence of enjoying a particular status during the interval and in the prescribed location e.g., retrieval access on a university document within the formation on financial deals may be assigned to the registrar with
preferred status during his tenure as registrar. Any number of (physical and atomic) actions can be accommodated in specifications of STASAC policies.

Status level is a named position of an agent that is relative to other status levels of interest in a specific domain of discourse. For example, a registrar and ordinary professor may be two distinct status levels in a university environment, with the registrar status being a higher level status than an ordinary professor. An agent’s status is determined from the agent’s ascribed status or action status at a particular time and location. An organization, with its formation rules are protected by a STASAC policy. The information about the organizations access policies are publicly accessible so that agents can choose to act in ways to increase their status and thus their access privileges on objects. The notion of an event is of primary importance in this proposed spatio-temporal action status access control model. The importance of the concept of events is known from their widespread use in the areas of linguistics and knowledge representation. Moreover, events provide a categorically homogeneous basis for representing a feature that is an essential aspect of this model. In this model, events are happenings during time interval \((t_1, t_2)\). That is, the model of time that is chosen in this work is a total ordering of time intervals that is isomorphic to the natural numbers. In this work, times are represented by natural numbers in YYYYMMDD format. Moreover, it is assumed that time is bidirectional so that proactive and post active changes may be made to represent access policy requirements. In time stamping past, present, future times and location of an agent are used as separate attributes in the model to make access control model as tuple stamping model. An agents access level is varied by evaluating queries (i.e. perform retrieval requests) on information sources which are protected by STASAC policies.

The assignment of agents and access privileges on objects changes dynamically with respect to change in time and location and as a consequence, the occurrences of events are ordered in an application being
modeled. Moreover, the situational factors, such as the time during which access to a resource is requested and the location of the agent requesting access are obtained from the monitoring agent and then the access privileges are modified.

5.6.2 Event description

Definition 16 A security event description is a finite set of 2-place assertions, that describe an event, when happens during time interval \((t_1, t_2)\) and at location \((l_n, l_a)\) identified by \(e_k\), the \(k \in \mathbb{N}\) and which includes three necessary facts and \(n(n \geq 0)\) non-necessary facts. A necessary facts that must appear in security even description \(e\) and non-necessary facts must appear in application specific security event description SED.

Three necessary facts in SED and its meanings are as follows:

- \(\text{happens } (e_k, t_1, t_2, l_n, l_a)\) - the event \(e_k\) happens during the time interval \((t_1, t_2)\) at location \((l_n, l_a)\)

- \(\text{act } (e_k, a_k, t_1, t_2, l_n, l_a)\) - the event \(e_k\) involves an action \(a_k\) at location \((l_n, l_a)\) during the time interval \((t_1, t_2)\)

- \(\text{agent } (e_k, u_k, t_1, t_2, l_n, l_a)\) - the event \(e_k\) the agent \(u_k\), time interval \(T\) and location the user where the event happens

It follows from the above facts is that happens \((e_k, t_1, t_2, l_n, l_a)\), act \((e_k, a_k, t_1, t_2, l_n, l_a)\) and agent \((e_k, u_k, t_1, t_2, l_n, l_a)\) are used represent a happening event \(e_k\) during time interval agent \((t_1, t_2)\) of action act \(a_k\) performance by an user agent \(u_k\).
5.6.3 Status Level Assignment

The set of axioms that specify the assignment of an agent to a status level is described in this section as follows:

An user agent $u_k$ is currently assigned the status level $l_1$ if an event $e_1$ happened during time interval $(t_1,t_2)$, that is earlier than the current time $T$, and resulted in the initiation of user agent $u_k$’s assignment to status level $l_1$ and this assignment has not been ended before $T$ as a consequence of an event $e_2$ happening during the time interval $(t_1,t_3)$ that cause user agent $u_k$’s assignment of this status level $l_1$ to be terminated.

The following are the two rules designed for status level assignment is STASAC model. Here we use 'asl' for status level assignment.

**Rule R$_1$**

$$\text{asl} (u_k, l_1) \leftarrow \text{current time} (T), \text{agent} (e_1,u_k),$$

$$\text{happens} (e_1, (t_1, t_2), l_n, l_a), \text{act} (e_1, a_k), (t_1, t_2) \leq T, (l_{n1}, l_{a1}) \in L$$

$$\text{asl} \_\text{init} (e_1, u_k, a_k, l_1, (t_1, t_2), T, l_{n1}, l_{a1}),$$

$$\text{not ended} \text{asl} (u_k, l_1, (t_1, t_2), T, l_{n1}, l_{a1}).$$

**Rule R$_2$**

$$\text{ended asl} (u_k, l_1, (t_1, t_2), l_{n1}, l_{a1}, T) \leftarrow \text{agent} (e_2, u_k),$$

$$\text{happens} (e_2, (t_3, t_4), l_n, l_a),$$

$$\text{act} (e_2, a_k),$$

$$\text{asl} \_\text{term} (e_2, u_k, a_k, (t_1, t_2), l_{n2}, l_{a2}), (t_1, t_2) \leq (t_3, t_4),$$

$$(t_3, t_4) \leq T, (l_{n2}, l_{a2}) \in L$$

(Location of the agent)

The above to rules R$_1$ and R$_2$ are core rules of the STASAC model.
5.6.4 Permission Assignment

In addition to assigning of status level to be user agent, access privileges on objects (i.e., permissions) are assigned to a status level, that permissions are denied to agents with a certain status.

The permission assignment and permission denials are described as follows:

- \( \text{pa}(p, \text{obj}, k_i, \text{sl}_i, (t_1, t_2), (n, a)) \) iff the access privilege \( p \) on the resource object \( \text{obj} \) by the user agent resource identified \( k_i \) to a status level \( \text{sl}_i \) when it satisfies spatio-temporal constrains \( (t_1, t_2) \) and \( (n, a) \).

- \( \text{pd}(p, \text{obj}, k_i, \text{sl}_i, (t_1, t_2), (n, a)) \) the access privilege \( p \) on the resource object \( \text{obj} \) is denied to the status level \( \text{sl}_i \) during the time interval \( (t_1, t_2) \) at the location \( (n, a) \).

**Definition 17**

In the STASAC model, the authorizations are defined as follows:

Authorization is 5 tuple \((\text{U, a, obj, (t_1, t_2), (n, a)})\),

where

- \( \text{U} \) is set of user agent,
- \( \text{a} \) is an access request to resources object \( \text{obj} \)
- \( \text{obj} \) is the resources object that user requests and \( (t_1, t_2) \) is time interval during which the user agent is trying to access an object and \( (n, a) \) is location from which user agent is trying for an access an resources.

It follows from above definition is that on receiving an access request from an user agent \( \text{U} \), the request is evaluated as follows:
Step 1: Load user(U) profile(s) (ie all the information known about the user agent including past history)

Step 2: Determine user agent's current status level when it satisfies spatio-temporal constraints

Step 3: Determine the spatio-temporal access policies that apply to user agent with the status level.

Step 4: Check whether access on object \( \text{obj} \in O \) as defined is assigned to users

Step 5: If user U has sufficient stats level \( \text{l} \in L \) and satisfies spatio-temporal constraints then the user request is permitted to access the resource.

5.7 IMPLEMENTATION OF SPATIO-TEMPORAL ACTION STATUS ACCESS CONTROL

In this research work, a spatio-temporal action status access control has been implemented using intelligent agents for various functionalities. Initially the system obtains a user authentication details when they try to enter the system. Now the rule manager checks the privileges of the user with the rule base where the details of the all users have been stored with their access rights. The permissions of the users are validated with the access rights and in addition, the spatio-temporal constraints along with the status level of the user are used for validating spatio-temporal action status access control policies. The following algorithm describes the method followed in this work.
Algorithm:

Step1 : Initialize Database

Step2 : Obtain the user query

Step3 : Check (user’s status level)

Step4 : Check for the access rights using the time, location and status level of the user

Step5 : Analyse history for status level <0.5 by applying the following rule:

If status level >0.5 then

Permit (access, object, \( t_1, t_2, \ell_n, \ell_a, \text{status level} \))

Modify access level

Else

Store the user details

Deny the access,

Step6: Check for next request,

Step7: Perform database operation

Table 5.4 shows the number of users requests prevented which violations prevented by intelligent temporal access control model.
Table 5.4  The Number of User Requests Denied Access by ASAC and STASAC Model

<table>
<thead>
<tr>
<th>Exp.No</th>
<th>No.of User Request Tried</th>
<th>No.of requests denied by ASAC</th>
<th>No.of requests denied by STASAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp1</td>
<td>100</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>Exp2</td>
<td>200</td>
<td>11</td>
<td>16</td>
</tr>
<tr>
<td>Exp3</td>
<td>300</td>
<td>18</td>
<td>22</td>
</tr>
<tr>
<td>Exp4</td>
<td>400</td>
<td>25</td>
<td>30</td>
</tr>
<tr>
<td>Exp5</td>
<td>500</td>
<td>34</td>
<td>39</td>
</tr>
</tbody>
</table>

From this table, it can be observed that STASAC model provides more security when comparing with Action Status Access Control Model.

5.8 PRIVACY AWARE ROLE BASED ACCESS CONTROL

Privacy-Aware Role Based Access Control (P-RBAC) model is used to enforce access control to data containing personally identifiable information and as such privacy sensitive. In this model, privacy policies are designed as permission assignments. Because of the additional components like purpose binding, condition and obligations introduced in the model, the permission assignments are also differs. However, during permission assignments (PA) conflict may occur. These conflicts can be detected using conflict detection algorithm. Moreover, as the conflict detection algorithm proposed by Qun Ni (2010) can detect conflicts only upto two permission assignments. We have extended this algorithm to detect the conflicts upto n permission assignments.

5.9 MODIFIED CONFLICTS DETECTION ALGORITHM

In this research work, a P-RBAC has been implemented using modified conflict detection algorithm for multiple permission assignments. The algorithm described as follows.
Algorithm:

PA – Permission Assignment, Pacl – Array list of conflicting permissions
Pal – Array list of all permission assignments already made

1. \[ \text{result} \leftarrow \text{Condition Validity Test (PA.condition, cv1, cv2, \ldots, cvn)} \]
2. if result = -1 then
3.   exit // invalid condition
4. end if
5. for all pa such that pa ∈ pal do
6.   for i = 1 to n do
7.     \[ \text{result} \leftarrow \text{Condition Conflict Test (PA.condition, pa[i].condition, cv[i].condition)} \]
8.   if its result = -1 then
9.     do begin
10.    for j = 1 to n do
11.      pacl.add (pa[j], cv[j]) // conflicting permission
12.     end
13.   exit
14. end if
15. end for
16. for i = 1 to n do
17.   \[ \text{result} \leftarrow \text{Obligation Ambiguity Test (PA.obligation, pa[i].obligation, cv[i].obligation)} \]
18. if its result = -1 then
19.   do begin
20.     for j = 1 to n do
21.       pacl.add (pa[j], cv[j])
22.     end
23.   exit
24. end if
25. end for
26. if PA.purpose ≠ PA.purpose. intended then
27.   pacl.add (pa[i].result)
28. end if
29. if result equals to 1 then
30.   assg.add (PA, CV)
31. end if
32. end for
5.10 RESULTS AND DISCUSSION

The Multiple-PA-Conflict Detection algorithm takes the requested permission as input and divides it into the atomic level. At this level the entries are checked with the already existing values in the previous permission assignments. The Condition-Validity-Test algorithm, Condition-Conflict-Test algorithm, Obligation-Ambiguity-Test algorithms are used to find if a conflict occurs. Further, the purpose is also checked so that it matches the intended purpose of the data. If at any of the stage a conflict occurs, they are noted and a detailed report is given indicating where the conflict occurs and also with which permission assignment the conflict occurs. By providing such detailed reports the user and the administrator can make use of it to avoid the conflict and revise any of the existing permission assignment.

Figure 5.5 Conflicts Detection Accuracy Analysis

Figure 5.5 shows that the modified multiple conflict detection algorithms have the capability to check the conflicts that occur in the three components namely purpose, obligation and condition. The Conflict Detection Algorithm can spot conflict only between two permissions. But the modified algorithm overcomes this and can detect conflicts in two or more permission assignments.
5.11 CONCLUSION

In this work, two new access control techniques namely an intelligent spatio-temporal role based access control model and a spatio-temporal action status access control model have been proposed. In addition privacy is considered in providing access control. New techniques have been proposed using rules, constraints and agents in this work in order to enhance the security of the database system.