Chapter 7

7. Extending the framework for Policy-conflict resolution

Policy conflicts arise due to separate set of attribute values or constraint such that no feasible state of the system exist that is compliant with the merged policies. The approach for policy conflict analysis entails analyzing given set of policies for potential conflicts and devising algorithms that could dynamically resolve the identified conflicts. The easiest would have been if the patient had personal web account storing entire medical information allowing the patient to define the access of these records as per his/her choice. But then medical science is complex and beyond the understanding of a common man. Health organizations comprise of care-providers with varied capabilities to handle patient’s sensitive health data. Hence, what should be disclosed to whom requires the involvement of medical professional. The authority of these professionals not only depends on the qualifications, positions, experience but also the facilities and size of the organization. Thus, people with similar qualification and experience working in different hospitals might differ in positions and responsibilities.

Conflicts can be detected by pair-wise comparison of all rules in the rule set of ACPs. The proposed framework proves to identify prevalent policy-conflicts and enables rule reduction on merging of two or more policies. The verification concludes that such conflicts are traceable and justify the robustness of the proposed framework. The challenge lies in composition of a consistent and conflict-free interoperation policy that governs all the inter-domain information and resource exchange. The objective of this chapter is to develop algorithms that could enhance the robustness of the proposed framework enabling it to handle and resolve maximum policy-conflicts detected on merging of ACPs of disparate healthcare organizations.

7.1 Causes of Conflicts

Verification of proposed framework revealed possible causes leading to policy conflicts. The causes identified forms the basis of further devising a solution to mitigate and avoid the emergence of these conflicts in merging of distant access control policies.

- **Missing hierarchies:** Access control policies are defined as per the user roles in the organization hierarchy. Healthcare organizations do not follow any set standard to
formulate these hierarchies. With the disparity in healthcare setups, the volume of users and level of roles vary broadly. Also, the user with the same role may have different responsibilities and tasks assigned by the respective administration. Policy-conflicts arise with the existence of difference in the user hierarchies of two organizations merged for sharing the required data.

- **Modality conflicts**: A rule is a subset of some other rule or value of an attribute of a rule contradicts with the similar attribute in other rule. Access policies laid by every organization vary with respect to rights and privileges assigned to the users. A possibility exists where certain right assigned to a role in one organization contradict with similar role in other organization. Assumingly, organization A allows the ‘visiting doctor’ role to read and write into patient’s diagnostic data whereas organization B allows the ‘visiting doctor’ role only to read the patient’s diagnostic data. Also, reading the past history of patient is considered as a separate role in A but is implicitly included in diagnostic data in B.

- **Missing attributes**: An attribute in one rule is undefined thus restricting its merging with other rule and resulting in conflicting rules. As stated, every organization defines their own set of permissions and privileges on the data. It is possible that a right or a privilege available in one organization may be not available in another organization. For ex., printing of diagnostic details of the patient might be permitted in organization A and no such privilege may exist in organization B. Merging of such rules would not be able to reach to a concrete decision of permitting or denying printing of diagnostic data in such a scenario.

- **Contradictory authorizations**: The authorizations obtained contradict where a higher authorization requires permissions from the lower level authorization in two disparate rule-sets. Authorizations are determined with the position acquired by the user in organization hierarchy. The user at the top level in the hierarchy has higher authority to access and further grant access to other users positioned at lower-levels. When two policies of disparate organizations are merged, a scenario may exist where the authorization set by organization A stands at a lower level in organization B, thus resulting in contradiction where a lower-level user is granting permission to a higher-level user in two different hierarchies.

- **Non-matching authorizations**: The authorizations set in rule-policies of disparate organizations display distinct values unable to provide clear decisions of allowing/denying access to the required data. It is highly challenging to resolve this disparity as
authorizations are considered as security controls restricting unauthorized access to the data. Giving preference and setting priority for acceptable authorization dynamically is difficult to determine.

The purpose of this study is to resolve the access control policy conflicts emerging on merging of two or more policies of disparate hospitals. The conflicts are categorized on rule attributes and their attribute values. The policies considered for this study are of health providers seeking to share the EHR of a patient in interoperable healthcare environment.

### 7.2 Categorization of Policy Conflicts

Policy conflicts are categorized differently by various researchers [200][201][202][203] while proposing conflict-resolution techniques in multi-domain data sharing environments.

**User Conflict:** The Subject (S) attribute in each rule defines the user to whom the access to the respective Resource (R) be permitted or denied. The condition where the Subject attribute in both the rules either contain different values or result in missing values raises user conflicts.

**Resource Conflict:** The Resource (R) attribute holds the data i.e. EHR which is further categorized according to the type of data stored by different users. The condition where the resource (EHR) attribute in both the rules either contain different values or result in missing values raises resource conflicts.

**Privilege Conflict:** The Action (A) attribute defines the rights and privileges for permitting or denying the Subject’s access on the listed Resource. The condition where the action (read, write etc.) contradicts between rules result in privilege conflicts.

**Constraint Conflict:** XACML provides an optional condition (C) attribute that is used to impose the conditional constraint on access the resource. The condition is defined as an explicit control that must be met to perform the stated action in the rule. The condition where conditional constraints are either undefined or contradicts with other rules in the policies result in constraint conflicts.

**Security Level Conflict:** HSA generates Security Level (SL) for each user (SL_U) attribute specified in the rule. The SL determines the positional hierarchy of the user after the policies of two disparate health organizations are merged. It indicates the level of access permissible
to the user. The difference in SL values needs to be addressed specifically when the user with lower SL wants to access data that the user with higher SL can only access. Similar implications have to make when the difference in SL values assigned to resource (SL_R) is identified.

**Authorization Conflict:** Authorization (Auth) is a subset of Subject Attribute and defines the user who would be responsible in allowing or denying access to the Resource. The condition where authorizations are either undefined or show distinct values in the rules of both policies result in authorization conflicts.

### 7.3 Policy-conflict Resolution Techniques

Reeder et al. [200] proposed a conflict-resolution technique based on specificity precedence i.e. a rule that defines more specific entity (Subject or Resource) takes precedence over the rule that specifies a more general entity. For ex., if a resource defined in Rule A states ‘diagnostic details’ of the patient and the resource defined in Rule B is ‘test advised’ then Rule A takes precedence over Rule B on the pretext that ‘test advised’ is a sub-data under ‘diagnostic details’. A similar approach is discussed by Nicole et al. [204] in addition to the discovery of several other techniques for conflict resolution.

One of the measure than can be taken is to assign distinct priority to each conflicting policy and allow the system to execute policy with higher priority. It is not suitable where the policies are arranged in hierarchies and conflict may occur differently at different levels in the hierarchy. Jatinder et al. [201] built a messaging system for customizing the information as per the real-world needs of health professionals. The middleware model works at a local level of administration in healthcare domain and claim to support data interoperability within the published framework. The major contributors in resolving the policy-conflicts revolve around recipient’s credentials, message content and environmental context. Charalambides et al. [202] proposed an approach to analyse, detect and dynamically resolve policy conflicts using ‘Identify-classify-detect-resolve’ principle in DiffServ Quality of Service (QoS) management domain. Its implementation uses Event Calculus that identifies the conflict and the reason why that conflict would have occurred.

The model proposed by Imine et al. [216] deals with the latencies and resolves the policy conflict changes dynamically but allows any change to existing data-sets only at the administrator level. The users play a vital role in providing health services to the patient.
Each user having any position in the organizational structure is equally crucial in the service-chain system of the organization. The organization experience frequent role-reversal of its users as per the demand and schedule. A domain-based approach [217] hides the sensitive data thereby acting as a filter at the object level. It establishes a security level on the domain of actors dynamically created and modified as per the need. Some organizations require reviewing the capabilities associated with subjects and their attributes defined through the access control policies. Access Control Lists [164] work on the attribute of “identity” whereas RBAC works on the attribute of “role”. The policies in ABAC expresses a complex Boolean rule set capable of evaluating many different attributes. The RBAC-based model [207] captures the disparities of access control policies relying on specific constraint-checks to administer interoperability. The constraints in health domain are subject to change [218] with the dynamic creation of demand. It has to handle various contingencies where the system needs to change its behavior and normal course of actions.

The above discussion is a clear indication that identification of users and further finding the precedence and specificity of their request is a prerequisite in resolving policy conflicts in interoperable environments. Conflict resolution can be computationally expensive. Hence, an attempt should be made to resolve the conflicts statically. Moreover, it is feasible to predict the resolution technique or solution statically as opposed to dynamic resolution as it is unpredictable and uncertain in determining the required or demanded solution. Also, not all conflicts can be resolved and possibility of detection of such conflicts is higher in static environment. This approach would drastically reduce the volume of probable conflicts to be resolved dynamically.

7.4 Integrating Conflict Resolution in Proposed Framework

Policy-conflicts identified on verification of the HSA-refined policy sets are resolved by logically fine-graining the conflicting rules further. The modified rules are fed to the HSA to be re-evaluated and merged, thus, providing clear decision to access the data. As shown in figure 7.1, a new component “change of policies using conflict resolution techniques” is added to the proposed framework that detects the type of conflict raised and accordingly calls the appropriate routine for its resolution.
Integration and merging policy sets of disparate healthcare organizations resulted in matching and conflicting rules. The matching rules exhibit the similarities in disparate set of policies and indicate a positive mapping between them. The conflicting rules exhibit dissimilarities in the policy sets and hence are rejected for obtaining final decision of permitting or denying access to the specified data. These rejected roles are then fed to the newly added component to be resolved such that an authorized relationship can be established in such rules, thus, increasing the chances of data availability without compromising security in interoperable healthcare environment.

Figure 7.1: Integration of Policy Conflict Resolution in HSA
The policy sets \{P_1,...,P_n\} and \{P_1,...,P_m\} of respective hospitals collaborate to share their data. The policies are inputted to HSA that generates the set of similar \{R_i,...,R_l\} and dissimilar \{R_i,...,R_k\} rules. All similar rules generate clear permit/deny decisions. All dissimilar rules are then checked to identify the attribute causing the conflict. Accordingly, the applicable routine is called and the rules are modified. Figure 7.2 expands the newly added component to the proposed framework. HSA detects the policy conflicts in the form of dissimilar rules where the rules are further analysed to detect the type of conflict. The appropriate algorithm is then executed to obtain the decision for resolving the identified conflicts. If required, the policies are fine-grained as per the acceptable decision and are referred to HSA.

7.5 Conflict-Resolution Algorithms

Following algorithms are proposed that not only resolve these conflicts but also work as a feedback system for obtaining future predictions and decisions on existence of similar situations. Each conflict being resolved is simultaneously recorded in a database of the concerned healthcare system as log. This log can be referred by the system administrator for monitoring and resolving any future conflicts of similar nature. Further, this repository of
transactions sets a timestamp bounding the user to gain session-based access to the otherwise unauthorized data. The algorithms handle the conflicting roles of particular type only. Algorithm 1 can handle the user, action (privilege) and conditional constraint conflicts. Algorithm 2 handles user’s security level and authorization conflicts whereas algorithm 3 specifies the solution to resolve resource’s security level conflicts. Each algorithm takes the conflicting rule as input and generates a feasible solution. Further, the resolution also ensures not to tamper with the original definitions of the access control policies. Hence, a view is created to permit access on the data as per the agreed authorization. Simultaneously, a log entry of this modification is recorded for future monitoring and performance checks.

**Algorithm 1: User, Privilege or Constraint conflicts**

Algorithm 1 is devised to resolves user, privilege and constraints or conditional policy conflicts. User conflicts can be understood as the discrepancy in Subject attribute of two rules where either the attribute values differ or are missing in any one rule-set of distinct policies. Privileges are the actions a user can perform if the access is granted on the specified resource. Missing or non-matching attribute values in the Action attribute results in privilege conflicts. As stated, conditions are optional and set as per the sole decision of each organization. Hence, unclear or unstated conditions in either of the rule set results in constraint conflicts.

To resolve user conflicts, the conflicting user’s position in their respective hierarchies is obtained. The user with the higher position in the hierarchy is given authorization and action is set accordingly. In case, of privilege and conditional conflicts, existing user position is compared and authorization is set with the user higher than the current user position identified in the conflicting rule. The newly set authorizations need to provide a mutual consent for setting the required action thus, resolving the raised conflict. The pseudocode to implement algorithm 1 is stated below.

1. Obtain the user position in their respective hierarchies
2. Compare the positions and set authorization to the user with higher position
3. Check if positions are same then
4. Increment user position by 1 if user position > maximum hierarchy level
5. Set authorization to incremented user position
6. Else Obtain consent from incremented positions
7. Set authorization to obtained consent

**Algorithm 2: Authorization or SL conflicts**

Algorithm 2 resolves Authorization and Security Level (SL) conflicts. Security levels are the end-product of HSA that assigns respective SL to each user and proves to handle secured interoperation of data in healthcare organizations. The SLs also contributes to policy conflicts by displaying differences in SLs of the users wishing to collaborate and share data. Authorization is another end-product of HSA that defines the authority over the Subject attribute and ensures the access only under the controlled supervision of the defined authority. Authorizations are set wherever deemed necessary by the health administrators. Merging of distinct policies generate conflicts of this type if the authorizations are found missing or are different leading to contradictions.

Algorithm 2 resolves SL and authorization conflicts by verifying the user position in the organization hierarchy and setting authorization of the user that is one level up from the user defined in the conflicting role. To illustrate, if SL in P1 is lower in its hierarchy but higher than SL in P2 then set authorization to one level up in P1. If authorization is set to maximum hierarchy level then leave the conflict unresolved.

/* If no authorization is set in some rule then Select user with highest SL */

1. Obtain the user position in their respective hierarchies
2. Check and compare the SL assigned to the users
3. Select the Action of the role where SL is found higher than the other
4. Otherwise, increment the user position and select its SL
5. Set authorization to the obtained SL in the respective role
6. if the conflicting SL is the highest in the user hierarchy then
7. display message “Cannot be Resolved”

**Algorithm 3: for resource conflict**

Algorithm 3 resolves the resource conflicts. RL assigned to the data in the EHR hierarchy limits the contents accessible to the defined users. Resource e.g. diagnostic details may have different RLs according to the confidentiality required for the specific patient. Conflict in RL is seen as the difference in RLs of the rules under verification. The conflict is resolved by giving weight-age to the policy whose resource level (RL) is lower than the other. Algorithm 3 resolves RL conflicts simply by selecting the role with lower RL value. Data being the core
of any application, highly crucial in particular to healthcare domain must be protected against any unwanted access. RLs categorized data between low and high with low requiring less protection as compared with high. Hence, in case of conflict, access to the lower RL is permitted to resolve the conflict.

/* If the conflict arise in resource level then Select rule with lowest EHR level */

1. Check the resource SL and select resource with lowest SL
2. Set authorization to lowest resource SL

7.6 Verification and Observations

The policy sets of two hospitals HA and HB are defined and fine-grained using HSA. The hierarchical distances calculated for user hierarchy of HA and HB is listed in table 7.1. As visible from table 7.1, both hospitals represent difference in their organizational structure with respect to the users’ positions and concerned authorities. HSA converts these values into unique SLs assigned with the subject attribute in the defined rules of ACPs of HA and HB.

Table 7.1: Hierarchical position obtained from the User Hierarchies of HA and HB.

<table>
<thead>
<tr>
<th>Users</th>
<th>Security Level (HA)</th>
<th>Security Level (HB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS</td>
<td>5</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>HOD</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Family Doctor</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Specialist</td>
<td>3</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>Patient</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Nurse</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Medical Technician</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Administrative Staff</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 7.2 lists the security levels of users generated after executing HSA. It represents the users of HA and HB permitted to share each other’s EHR systems if the query containing the Security Level of User (SL_U) matches with these values. For ex., the doctor (Family_Doc) of HA and HB are permitted to share the data if both exhibit SL_U equal to 4 whereas if the role in HA is of Family_Doc and in HB the role is of Nurse, and then SL_U must be set to 1. A similar matrix is generated for obtaining the security level for resources (SL_R) that is then mapped accordingly with each SL_U while devising rules in ACPs of subsequent users.

Table 7.2: Security Level (SL_U) for the users of HA and HB calculated by HSA

| Users            | HA | HB
|------------------|----|----|
|                  |    | Family_Doc
|                  |    | Med_Tech
|                  |    | HOD
|                  |    | Specialist
|                  |    | Jr. Resident
|                  |    | Nurse
|                  |    | Admn_Tech |
Table 7.3 lists the requested queries on the policy sets of HA and HB. The queries are executed and verified with one to one attribute value of each rule in both the policy sets. The conflicts arising due to difference in utmost one attribute value are considered to be resolved applying appropriate resolving algorithm. Table 7.3 further identifies the type of conflict and provides the requirement and method of resolving the conflict, wherever applicable.

<table>
<thead>
<tr>
<th>Query ID</th>
<th>Request Query</th>
<th>Requester Policy</th>
<th>Sender Policy</th>
<th>Conflict type</th>
<th>Resolution Algorithm</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{Family_Doc, 1, Cl_diag, 2, write}</td>
<td>Effect is Permit</td>
<td>Effect is deny</td>
<td>Effect</td>
<td>None</td>
<td>Leave unresolved</td>
</tr>
<tr>
<td>2</td>
<td>{Med_tech, 1, Cl_diag, 2, write, Family_Doc}</td>
<td>Authorization is Family_Doc</td>
<td>Authorization is HOD</td>
<td>Authorizat</td>
<td>Algorithm 2</td>
<td>Set requester’s authorization = or &gt; the Authorization in the sender’s policy</td>
</tr>
<tr>
<td>3</td>
<td>{Specialist, 2, Cl_diag, 3, write}</td>
<td>SL_U is 2</td>
<td>SL_U is 1</td>
<td>SL_U</td>
<td>Algorithm 2</td>
<td>Set requester’s Authorization &gt; Hierarchy position of Specialist</td>
</tr>
<tr>
<td>4</td>
<td>{Family_Doc, 1, Cl_diag, 2, write}</td>
<td>SL_R is 2</td>
<td>SL_R is 1</td>
<td>SL_R</td>
<td>Algorithm 3</td>
<td>Set requester’s SL_R &lt; or = sender’s SL_R and set requester’s authorization &gt; Hierarchy position of Family_Doc</td>
</tr>
<tr>
<td>5</td>
<td>{Med_tech, 1, Cl_diag, 1, read}</td>
<td>Action is read</td>
<td>Action is write</td>
<td>Action</td>
<td>Algorithm 1</td>
<td>Identify the requester’s and sender’s Hierarchy position and increment them by 1. Also, set the obtained user’s as Authorization of both policies respectively, Obtain consent and set the Action attribute accordingly</td>
</tr>
</tbody>
</table>
Observations

Query (Query_Id = 1) in figure 7.3 reveals a contradiction in the Effect component of two policies where HA permits and HB denies access to the specified user. This conflict is left unresolved as it is impossible to provide a logical reasoning to select either permit or deny access decision in such case.

Authorization conflict can be understood to occur for two reasons. Possibly, the authorization exhibit difference of attribute value in the rules or it may remain undefined in any one policy. Execution of the request query 2 reveals former cause leading to authorization conflict and deterring the access to be granted. Figure 7.4 indicates that the policies of HA (requester hospital) defines Authorization and set its value to ‘Family_Doc’ whereas no such authorization is set in the policies of HB (sender hospital), thus resulting in policy conflict of type ‘Authorization’.

The aforesaid conflict is resolved by applying the solution devised in algorithm 2. The user in the subject attribute of the conflicting rule is inputted to obtain the hierarchical position of that user. Authorization attribute is upgraded to user one level higher to the inputted user’s position. The logical reasoning conferring to this decision is that the user at a higher hierarchical position withholds higher authority and control on the data. Obtaining the
consent of such user would ensure legitimate and permitted access to the data. This would resolve the discrepancy in the disparate policies and enhance quality of care by providing timely access to the required data.

Observing the execution of Query_Id = 4 reveals the conflict type related to resource attribute. The requested policy demanding access to the resource exhibit the SL_R of resource is higher in comparison with the SL_R of sender’s policy. Algorithm 3 applies a simple rule where the resource with lesser SL is set for sharing under the authorization of higher level hierarchy in the requester’s hierarchy chart. The argument behind this decision is that the resource i.e. EHR is highly sensitive and confidential. It needs to be protected against illegal and undue disclosure to third-party users. Therefore, conflicts in resource are handled by denying access to the data which is marked at higher security levels. Also, permitting access of lower level resource is bounded with the authorization set to the higher level user of requesting policy.

Medical technician (Med_Tech) (Query_Id=5) demands a read access on patient’s diagnostic data whereas the sender policy permits Med_Tech to write more contents on the requested data. Resolution of such type of conflicts requires user intervention to provide the decision or needs an algorithm that could set the priority on different actions as per their probability of vulnerabilities and security threats to exposure of the data. The latter would require quantifying the Action attribute values. The resolution algorithm 1 increments the Hierarchy position of users defined in requester’s and sender’s policy by 1. The designated users are set as Authorization attribute in the policies respectively. Thereafter, mutual consent is obtained to grant the access giving priority to the requesting query and Action attribute is set accordingly.

<table>
<thead>
<tr>
<th>Query_ID</th>
<th>User Position</th>
<th>User Position + 1</th>
<th>Conflict Resolved</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-</td>
<td>-</td>
<td>Leave unresolved</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>Set authorization HOD in HA</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
<td>Set authorization HOD in HA</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>3</td>
<td>Set authorization Family_Doc in HB</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>2</td>
<td>Set authorization Nurse in HA and HB and seek their consent</td>
</tr>
</tbody>
</table>

Table 7.4: Identifying the changes in User’s SL during conflict resolution

Table 7.4 reflects the changes made to desired attribute for resolving the detected conflicts. Handling the conflict in query 2 required an alteration of authorization to HOD in HA, thus inculcating similarities between HA and HB for this query to generate an authorized decision when executed again. It is observed in query 3 that the user (Specialist (SL_U =2)) in HA
requests write permission on the diagnostic data (SL_R = 3) in HB. The policy in HB permits the Specialist with SL_U set to 1 for the same request. The conflict is resolved by setting authorization higher than the specialist in HA and equalizing the SL_U of HA with HB. Another complex resolution is experienced in query 5. The conflict type ‘Action’ being a subjective attribute could not prioritize its values and set precedence over them. Thus, authorization were established in both the policies and mutual consent obtained from the higher authorizations to reach to the final decision and refine the rules accordingly.

Such changes reduce the robustness of policies and make the data vulnerable to leakages. Conflict resolution must handle this issue seriously. Hence, instead of modifying the original policies, modifications required to mitigate and resolve the conflicts are applied via views. The views generate the relevant rules to match the requested query. Views are further updated to fine-grain the rules and generate clear decisions when merged again to satisfy the requested query. The views can be permanent or temporary depending on the usage of the resource and also according to the security demands on that resource. Simultaneously, a log entry of this modification is recorded for future monitoring and performance checks.

### 7.7 Summarization

This chapter presented the policy-conflict resolution algorithms devised to mitigate the conflicts rose due to merging of access control policies in interoperable healthcare domain. An off-line static analysis was performed on the rules set of ABAC policies using SQL server. Various types of conflicts were identified and handled accordingly. The algorithms designed to resolve the obtained conflicts revolved around handling of single attribute at a time, i.e., the algorithms are capable to handle atomic conflicts administered on merging of two rules in disparate policy sets. Moreover, the algorithms are authorization-centric and are devised on the assumption that setting and determining authorizations would automatically ensure legalized and relevant access to the critical and sensitive resources only. The objective was to resolve maximum possible conflicts to enhance data availability without compromising confidentiality and integrity in distributed EHR-systems.

Another aspect that is given due consideration is to ensure resolutions should not affect the originality of policies and make them flexible, thus forfeiting the very purpose of imposing any access constraints on the data. Hence, session-based views are generated to fine-grain the policies and allow sharing of data otherwise restricted to access.
The algorithms in this paper lack to handle conflicts arising due to difference in multiple attributes of rules in the policies. Also, it is required to generate all permutation combinations of same conflict arising in all participating rules and policies that can be included as an extension to the devised algorithms.

Each ACP in the EHR-system has its own states of operation. To implement a control over the conflicting policy, it is required to know the current state of that policy and other policies that contradict with it. It is computationally expensive to know the current state of the system that exhibit frequent change in the attribute values of these policies. Also, it is difficult to track the order and time when that change occurred. In order to make the resource available for providing quality services to the patients anywhere, anytime, it becomes immensely difficult to ensure secured and legitimate disclosure of resource to the authorized users. Thus, conflicts needs to be handled appropriately as they result in modification of existing states of the attributes leading to gain undue access on otherwise prohibited data. The algorithms proposed in this chapter are devised giving due weightage to prevent illegitimate and undue disclosure of EHR and ensure seamless sharing of EHR in interoperable healthcare environment. The effort is to ensure timely availability of data enabling continuity of care and services by the healthcare community.