Chức năng đánh giá chính sách

1. Đặc tả lý thuyết đánh giá chính sách:

1.1. Phân tích chính sách:

Phân ra làm hai dạng: attribute type (at) và applicable constraint (ac)
Đối với các at sẽ lưu trữ loại thuộc tính sử dụng trong policy. Ví dụ: tên, tuổi, địa chỉ, …
Đối với các ac sẽ lưu trữ các điều kiện so sánh trong policy. Ví dụ: tuổi > 18, địa chỉ tại "tp Hồ Chí Minh".

Ta xét ví dụ sau:
Nội dung chính của policy này là cung cấp quyền truy cập cho tập tin "**conference_rc**" cho các nhóm đối tượng người dùng bao gồm ba vai trò {**admin, pc-chair, pc-member**}. Các quyền cho phép trên tập tin này bao gồm {**read, write**}. Với các yêu cầu ràng buộc như sau:

- Nhóm người dùng có vai trò là **admin** sẽ có toàn quyền {**read, write**} trên tập tin **conference_rc.**
- Nhóm người dùng có vai trò là **pc-chair** sẽ có quyền đọc {**read**} trên tập tin **conference_rc.**
- Nhóm người dùng có vai trò là **pc-member** chỉ có quyền đọc {**read**} trên tập tin **conference_rc** khi và chỉ khi đang trong cuộc họp (**isMeeting = true**).

Các thuộc tính khác của chính sách mẫu:

- Effect: "**Permit**"
- Combining algorithm: "first-applicable"
- Thể hiện chính sách: "Permit"

Policy biểu diễn dưới dạng XML như sau:

```
<PolicySet PolicySetId="RPSlist.0" PolicyCombiningAlgId =
"urn:oasis:names:tc:xacml: 1.0:policy-combining-algorithm:first-applicable">
   <Target>
      <Resources>
        <Resource>
          <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-
equal">
            <AttributeValue
            DataType="http://www.w3.org/2001/XMLSchema#string">confere
            nce_rc</AttributeValue>
            <ResourceAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:
            resource:resource-id"
            DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ResourceMatch>
        </Resource>
      </Resources>
    </Target>
   <PolicySet
   PolicySetId="RPSlist.0.0"PolicyCombiningAlgId="urn:oasis:names:tc:xacml
   :1.0:policy-combining-algorithm:first-applicable">
      <Target/>
```

```
<Policy PolicyId="RPSlist.0.0.0"
RuleCombiningAlgId="urn:oasis:names:tc:xacml: 1.0:rule-combining-
algorithm:first-applicable">
 <Target>
   <Subjects>
     <Subject>
      <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-
      equal">
        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
        admin</AttributeValue>
          <SubjectAttributeDesignator SubjectCategory="urn:oasis:names:tc:
          xacml:1.0:subject-category:access-subject" AttributeId="role"
          DataType="http://www.w3.org/ 2001/XMLSchema#string"/>
           </SubjectMatch>
     </Subject>
   </Subjects>
   <Actions>
     <Action>
      <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-
      equal">
         <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
         read</AttributeValue>
          <ActionAttributeDesignator
          AttributeId="urn:oasis:names:tc:xacml:1.0 :action:action-id"
          DataType="http://www.w3.org/2001/XMLSchema #string"/>
       </ActionMatch>
     </Action>
     <Action>
      <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-
      equal">
         <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
         write</AttributeValue>
         <ActionAttributeDesignator
         AttributeId="urn:oasis:names:tc:xacml:1.0: action:action-id"
         DataType="http://www.w3.org/2001/XMLSchema#string"/>
       </ActionMatch>
     </Action>
   </Actions>
 </Target>
 <Rule RuleId="RPSlist.0.0.0.r.1" Effect="Permit">
   <Target/>
 </Rule>
</Policy>
<Policy PolicyId="RPSlist.0.0.1"
RuleCombiningAlgId="urn:oasis:names:tc:xacml: 1.0:rule-combining-
algorithm:first-applicable">
 <Target>
   <Subjects>
     <Subject>
      <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-
      equal">
        <AttributeValue
        DataType="http://www.w3.org/2001/XMLSchema#string">pc-
        chair</AttributeValue>
          <SubjectAttributeDesignator
          SubjectCategory="urn:oasis:names:tc:xacml :1.0:subject-
          category:access-subject" AttributeId="role" DataType="ht
          tp://www.w3.org/2001/XMLSchema#string"/>
       </SubjectMatch>
     </Subject>
   </Subjects>
   <Actions>
     <Action>
      <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-
      equal">
         <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
         read</AttributeValue>
```

```
            <ActionAttributeDesignator
            AttributeId="urn:oasis:names:tc:xacml:1.0: action:action-id"
            DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ActionMatch>
        </Action>
      </Actions>
    </Target>
    <Rule RuleId="RPSlist.0.0.1.r.1" Effect="Permit">
      <Target/>
    </Rule>
  </Policy>
  <Policy PolicyId="RPSlist.0.0.2"
  RuleCombiningAlgId="urn:oasis:names:tc:xacml: 1.0:rule-combining-
  algorithm:first-applicable">
    <Target>
      <Subjects>
        <Subject>
          <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-
          equal">
            <AttributeValue
            DataType="http://www.w3.org/2001/XMLSchema#string">pc-
            member</AttributeValue>
            <SubjectAttributeDesignator
            SubjectCategory="urn:oasis:names:tc:xacml: 1.0:subject-
            category:access-subject"AttributeId="role"DataType="http:
            //www.w3.org/2001/XMLSchema#string"/>
          </SubjectMatch>
          <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-
          equal">
           <AttributeValue
           DataType="http://www.w3.org/2001/XMLSchema#string">true
           </AttributeValue>
            <SubjectAttributeDesignator
            SubjectCategory="urn:oasis:names:tc:xacml: 1.0:subject-
            category:access-subject" AttributeId="isMeeting"DataType=
            "http://www.w3.org/2001/XMLSchema#string"/>
          </SubjectMatch>
        </Subject>
      </Subjects>
      <Actions>
        <Action>
          <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-
          equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
            read</AttributeValue>
            <ActionAttributeDesignator
            AttributeId="urn:oasis:names:tc:xacml:1.0: action:action-id"
            DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ActionMatch>
        </Action>
      </Actions>
    </Target>
    <Rule RuleId="RPSlist.0.0.2.r.1" Effect="Permit">
      <Target/>
    </Rule>
  </Policy>
  <Policy PolicyId="RPSlist.0.0.3"
  RuleCombiningAlgId="urn:oasis:names:tc:xacml: 1.0:rule-combining-
  algorithm:first-applicable">
    <Target/>
    <Rule RuleId="RPSlist.0.0.3.r.1" Effect="Deny">
      <Target/>
    </Rule>
  </Policy>
 </PolicySet>
</PolicySet>
```

Đầu tiên ta sẽ lấy các kiểu thuộc tính (at) được sử dụng trong policy này:

Lưu ý: vì effect của policy là permit nên mình sẽ gán các at theo hướng permit. Thể hiện như sau:

{
at_0: $\forall v \in$ **Resource-type**
at_1: $\forall v \in$ **Role**
at_2: $\forall v \in$ **action-id**
at_3: $\forall v \in$ **isMeeting**
}

Ở đây các at không cần ghi rõ các giá trị cụ thể có thể xảy ra. Ví dụ trong trường hợp của at_1 chỉ cần thuộc kiểu Role (vai trò của người dùng là được). Chưa xét đến vai trò có gía trị cụ thể là gì!! Tương tự với các giá trị của at còn lại.

Việc tiếp theo chúng ta sẽ xét đến các điều kiện ràng buộc (ac). Chúng ta tạo các ac theo policy ví dụ trên như sau:

PolicySet (**Effect** là Permit)
{
Xác định vùng dữ liệu tương tác (**resource**) và gán giá trị cho ac_0.
ac_0: **conference_rc** ∈ resource-type
Nhóm người dùng có vai trò là **Admin** sẽ có toàn quyền trên tập tin **conference_rc**
PolicySet (**Effect** là) (1)
{
ac_1: Role = **admin**
ac_2: action-id = **read**
ac_3: action-id = **write**
}
Nhóm người dùng có vai trò là **pc-chair** sẽ có toàn quyền **read** trên tập tin **conference_rc**
Policy (**Effect** là Permit) (2)
{
ac_4: Role = **pc-chair**
ac_5: action-id = **read**
}
Nhóm người dùng có vai trò là **pc-member** sẽ có toàn quyền **read** trên tập tin **conference_rc** khi và chỉ khi **isMeeting** = "true"
Policy (**Effect** là Permit) (3)
{
ac_6: Role = **pc-member**
ac_7: isMeeting = "**true**"
ac_8: action-id = **read**
}
Ngoài ba điều kiện trên tất cả yêu cầu của người dùng đều bị "Deny"
Policy (**Effect** là **Deny**) (4)
{
ac_9: **Deny**
}
}

Policy ban đầu sẽ được biểu diễn bằng các ac và at như sau:

(1) : <D, P, IN, NA>
D : Empty

P : or (and (ac_0; ac_1; ac_3); and (ac_0; ac_1; ac_2))
IN: or (not(at_0); not(at_1); not(at_2))
NA: not (ac_0)

Giải thích: ở Policy này dành cho đối tượng là **admin** vì có **effect** là **Permit** nên giá trị Deny là empty. Giá trị Not Applicable được trả về khi không thoả mãn ac_0. Giá trị Indeterminate được trả về khi không tồn tại bất kỳ một trong ba thuộc tính là: Resource-Type, Role và action-id. Cuối cùng xét đến trường hợp Permit khi thoả mãn tất cả 3 điều kiện là: **Admin** muốn **read** trên dữ liệu **conference_rc** hoặc **Admin** muốn **write** trên dữ liệu **conference_rc.**

(2) : <D, P, IN, NA>
D : Empty
P : and (ac_0; ac_4; ac_5)
IN: or (not(at_0); not(at_1); not(at_2))
NA: not (ac_0)

Giải thích: ở Policy này dành cho đối tượng là **pc-chair** vì có **effect** là **Permit** nên giá trị Deny là empty. Giá trị Not Applicable được trả về khi không thoả mãn ac_0. Giá trị Indeterminate được trả về khi không tồn tại bất kỳ một trong ba thuộc tính là: Resource-Type, Role và action-id. Cuối cùng xét đến trường hợp Permit khi thoả tất cả 3 điệu kiện là: **pc-chair** muốn **read** trên dữ liệu **conference_rc**

(3) : <D, P, IN, NA>
D : Empty
P : and (ac_0; ac_6; ac_7; ac_8)
IN: or (not(at_0); not(at_1); not(at_2); not(at_3))
NA: not (ac_0)

Giải thích: ở Policy này dành cho đối tượng là **pc-member** vì có **effect** là **Permit** nên giá trị Deny là empty. Giá trị Not Applicable được trả về khi không thoả mãn ac_0. Giá trị Indeterminate được trả về khi không tồn tại bất kỳ một trong bốn thuộc tính là: Resource-Type, Role, action-id và isMeeting. Cuối cùng xét đến trường hợp Permit khi thoả tất cả 4 điệu kiện là: **pc-member** muốn **read** trên dữ liệu **conference_rc** nếu đang có mặt ở **isMeeting**.

(4) : <D, P, IN, NA>
D : ac_9
P : Empty
IN: Empty
NA: not (ac_0)

Giải thích: luôn luôn trả về **true** vì effect có gía trị là Deny. Giá trị Not Applicable được trả về khi không thoả mãn ac_0.

Ở trên là quá trình phân tích policy theo dạng at và ac. Dưới đây là cách đánh giá chính sách và các dạng thuộc tính được hỗ trợ trong đánh giá chính sách:

1.2. Đánh giá chính sách:

Giả sử mình có các câu request như sau:

1. **Admin** muốn **đọc** file **conference_rc**
2. **Pc-chair** muốn **ghi** file **conference_rc**
3. **Pc-member** muốn **ghi** file **conference_rc** khi họ đang ở phòng họp (**isMeeting = "true"**).

Idea của việc đánh giá chính sách sẽ là: bất kể mỗi policy hoặc policy set nào đều có 4 vùng không gian quyết định (Decision Space) bao gồm Permit, Deny, Indeterminate và Not Applicable. Để biết được giá trị nào của policy sẽ được gán cho câu request mình cần xét xem trong 4 vùng không gian quyết định DS thì vùng nào trả về kết quả "**true**".
Xét ví dụ ở câu request 1:
Role = Admin
action-id = read
Resource-type = conference_rc

Phép toán được sử dụng trong quá trình đánh giá chính sách này là phép so sánh chuỗi.

Các thông tin trên trả về giá trị ac_0; ac_1; ac_2 = true (hay nói cách khác các giá trị thuộc tính trên đều thuộc lớp **positive**)

- ⇨ Vùng không gian Permit của Policy 1 sẽ trả về true: or (and (ac_0; ac_1; ac_3); and (ac_0; ac_1; ac_2)). Cả 3 vùng không gian còn lại đều trả về kết quả là "false"

Thứ tự ưu tiên đánh giá có thể đánh giá các vùng không gian từ Not Applicable, Indeterminate, Deny, Permit. Để làm tăng tính performance

Tương tự với yêu cầu thứ hai:
Role = Pc-chair
action-id = **write**
Resource-type = conference_rc

Các thông tin trên trả về giá trị ac_0; ac_4 = true (hay nói cách khác các giá trị thuộc tính trên đều thuộc lớp **positive**); riêng ac_5 = false (hay nói cách khác các giá trị thuộc tính trên đều thuộc lớp **negative**). **Lưu ý**: không thể đánh giá thông qua chính sách số 2 được vì không có vùng không gian nào trả về là true, tương tự như vậy với chính sách số 3 và kết quả cuối cùng là chính sách số 4; với giá trị Deny sẽ được trả về (ac_9).

Yêu cầu thứ 3
Role = Pc-member
action-id = **write**
Resource-type = conference_rc
isMeeting = true

Tương tự như yêu cầu thứ 2 kết quả trả về vẫn là Deny.

2. Các kiểu thuộc tính và các phép toán hỗ trợ:

2.1. Kiểu thuộc tính hỗ trợ

Ngoài các kiểu cơ bản như string, number, date, … các chính sách còn hỗ trợ thuộc tính có dạng dữ liệu phân cấp hoặc lấy dữ liệu từ CSDL.

Ví dụ: CanThoNationalHospital/HeartCenter/Hypertension.
Đối với dạng dữ liệu này phải xét theo thứ tự từ trái sang, nếu sai bất kỳ ở thành phần nào để trả về giá trị "false".
Hoặc dữ liệu dạng lấy giá trị cụ thể từ mức Collection đến fields. sẽ phải lấy giá trị của field cụ thể: PatientRecord/InformationRecord/Email = *VIP@gmail.com.

2.2. Các phép toán hỗ trợ

Các phép toán so sánh giá trị giữa chính sách và yêu cầu truy cập bao gồm:

- So sánh số học: <, >, =
  Ví dụ so sánh giữa hai gía trị số: Age > 18,
- So sánh chuỗi: ==, !=
  Ví dụ so sánh sự giống nhau giữa các chuỗi có trong chính sách và yêu cầu: Address = "Tp Hồ Chí Minh".
- So sánh trong tập hợp: $\subset, \supset, \in, \notin$
  Ví dụ so sánh gía trị của câu request có nằm trong khoảng của yêu cầu truy cập hay không: Age = [20; 50) hoặc email = {*@student*, *@e}

2.3. Dữ liệu dạng phân cấp

Đặc trưng chính trong mô hình này chính là dữ liệu dạng phân cấp. Có nghĩa là dữ liệu được lưu theo cấu trúc JSON (theo hướng tiếp cận trước đây của bài toán). Chính vì vậy việc hiện thực chức năng đánh giá chính sách phải thoả mãn được các yêu cầu so sánh theo ***path*** được định nghĩa trong Policy. Ta có ví dụ sau:

Policy:
{
    Role: Doctor;
    Position: CanThoNationalHospital/HeartCenter/Hypertension
}
Request:
{
    Role: Doctor;
    Position: Heart Attack
}
Lưu ý: Trong HeartCenter bao gồm hai khoa là Hypertension và Heart Attack.
Để so sánh thì phải vào đường dẫn cụ thể:
    Request: CanThoNationalHospital/HeartCenter/Heart Attack

2.4. Policy có trong hệ thống

Hệ thống sẽ có hai dạng policy bao gồm:

Policy được quản lý bởi admin: Policy này cho phép người dùng tương tác trực tiếp với vùng dữ liệu được bảo vệ nếu như thoả mãn các yêu cầu ràng buộc. Về mức độ ưu tiên thì thấp hơn policy do người quản trị dữ liệu quản lý.

Policy được quản lý bởi Data Owner: Policy này cho phép người chủ dữ liệu tạo ra để bảo vệ vùng dữ liệu của riêng họ. Policy này là yêu cầu bắt buộc phải thoả mãn vì vậy độ ưu tiên luôn là cao nhất.

Xét ví dụ về hai dạng policy trên như sau:
Administrator Policy: Cho phép đọc thông tin bệnh nhân nếu người đó là bác sĩ.
Data Owner Policy: Chỉ hiện toàn bộ thông tin về địa chỉ email và sđt cho bác sĩ chịu trách nhiệm chữa trị trực tiếp cho bệnh nhân đó. Trường hợp ngược lại chỉ hiện một phần và ẩn đi phần chi tiết.

Dữ liệu liên quan đến 2 policy đó như sau:
Hệ thống có hai bác sỹ là: Nguyễn Thị A và Lê Văn B
Danh sách bệnh nhân của bác sĩ Nguyễn Thị A gồm hai người:{Nguyễn văn Tèo, Lê Văn Tí}

Thông tin cá nhân của cả hai bệnh nhân đó như bên dưới:

```
{
        {
                Name: Nguyễn Văn Tèo;
                Age: 45;
                Địa chỉ: TP Cần Thơ;
                SĐT: 01234567899;
                Email:nvteo@gmail.com
        };
        {
                Name: Lê Văn Tí;
                Age: 40;
                Địa chỉ: TP Hồ Chí Minh;
                SĐT: 0901234567;
                Email: lvti@gmail.com
        }
}
```

Giả sử cả hai bác sĩ đều muốn truy xuất thông tin thì kết quả trả về như sau:
Nguyễn Thị A:

```
{
        {
                Name: Nguyễn Văn Tèo;
                Age: 45;
                Địa chỉ: TP Cần Thơ;
                SĐT: 01234567899;
                Email:nvteo@gmail.com
        };
        {
                Name: Lê Văn Tí;
```

Age: 40;
Địa chỉ: TP Hồ Chí Minh;
SĐT: 0901234567;
Email: lvti@gmail.com
}
}

Lê Văn B:
{
{
Name: Nguyễn Văn Tèo;
Age: 45;
Địa chỉ: TP Cần Thơ;
SĐT: 0123*;
Email: *@gmail.com
};
{
Name: Lê Văn Tí;
Age: 40;
Địa chỉ: TP Hồ Chí Minh;
SĐT: 090*;
Email: *@gmail.com
}
}