Vol.2018-CLE-25 No.7
2018/6/16

XACMLを用いた中継サーバの開発と共同指導における生徒情報閲覧制限の実現

堀 和彦[1,a)]　永井 孝幸[2,b)]

概要：教育現場のデジタル化が進められる中，電子化された情報の公開範囲は適切に管理される必要がある．児童生徒の成績や健康診断結果などの情報は，しかるべきユーザに対してのみ公開されることが望ましいが，それぞれの情報に対して個別に公開範囲を定義するのは難しい．本研究では，属性ベースのアクセスポリシー記述言語仕様である XACML を利用することで，データベースに格納された生徒と教師の「所属クラス」や「指導科目」などの属性をもとに，教育機関における個人情報の閲覧制限を実現する中継サーバを開発した．この中継サーバを用いることで共同指導における生徒情報の閲覧制限を実現する．

キーワード：XACML，ABAC，共同指導

Protecting student information under cooperative teaching environment by XACML-enabled proxy server

Hori Kazuhiko[1,a)]　Nagai Takayuki[2,b)]

Abstract: When information technology is introduced to educational organization, appropriate access control is necessary for digitalized information. Although it is desirable that information such as records of students and results of health checkups is disclosed only to appropriate users, it is difficult to define the scope of disclosure separately for each piece of information. In our research, we have developed an XACML-aware proxy server to restrict access to personal information stored in website of educational institution. The access control policy language XACML enalbes us to define attribute-based acccess control policies based on the relationship information between teachers and students managed in school database. As a testcase, we apply the proxy to restrict viewing of student information under cooperative teaching environment.

Keywords: XACML, ABAC, Cooperative teaching

1. はじめに

近年，文部科学省の主導のもとで教育の ICT 化が推進されており，日本政府は IT を活用した指導環境（IT に習熟した教員や学校教室の無線 LAN 導入）の整備を 2020 年度までに完了させることを目標にしている [1]．総務省が公開している「教育 ICT ガイドブック」[2] には，教育現場の ICT 化の代表的な先行事例として，クラウドを利用した学びの活性化・最適化を挙げている．教育機関における ICT 活用には個人情報の保護も合わせて考える必要がある．2017 年に文部科学省が策定した「教育情報セキュリティポリシーに関するガイドライン」[3] において，オンラインの情報に対して，教職員の業務負担軽減を考慮しつつ，適切な公開範囲を定める必要性が示されている．

現在広く用いられているアクセス制御方式に，**RBAC（Role-Based Access Control：ロールベースアクセス制御）**方式がある．RBAC 方式では，各人に与えられた特定のロール（役割）に対して，ある操作を実行するパーミッションを与える．ロールを通してパーミッションが与

1 京都工芸繊維大学，大学院工芸科学研究科情報工学専攻
2 京都工芸繊維大学，情報工学・人間科学系，准教授
a) k-hori18@dsm.cis.kit.ac.jp
b) nagai@kit.ac.jp

えられるため，アクセス権の管理が単純化されている．しかし，教育機関のデータのアクセス制御を行うには粒度が十分でない．例えば，教師のロールに「児童生徒の成績にアクセスする」許可を与えたとすると，教師は自分が担任していない児童生徒の成績を見る権利まで持ってしまうという問題が生じる．

RBAC の枠組みでこの問題を解決するために，「A クラスの教師・B クラスの教師」などにロールを細分化することは原理的に可能ではあるが，類似するロールが複数定義されることでルール定義が膨大になる'*role explosion*' が発生してしまう．したがって，教育機関のデータを管理・運用するためには，粒度が細かく，柔軟なルール定義が可能なアクセス制御の仕組みが必要になる．

このようなアクセス制御を実現する方式の一つに，**ABAC（Attribute-Based Access Control：属性ベースアクセス制御）**方式がある．ABAC 方式では，ユーザのロールに対してパーミッションを与えるのではなく，複数の属性（ユーザ属性，リソース属性，オブジェクト，環境属性など）を結合したポリシーを使用してユーザにパーミッションを与える．上で挙げた教師のロールの例では「所属クラス」という属性を教師と生徒に与えれば，1 つのルールで教師が閲覧できる児童生徒の情報を所属クラスの児童生徒のみに限定できるため，RBAC 方式で問題となる'*role explosion*' は発生しない．また，アクセス日時やデバイスの種類などの環境属性を利用できるため，ユーザの状態の変化に対して動的なアクセス制御を実現することができる．

そこで本研究では，児童生徒の成績や出席状況，健康診断結果などの情報（以下，**校務系情報**とする）を「直接指導している教師のみ閲覧できる」ように，また，児童生徒の宿題やワークシート，作品などの情報（以下，**学習系情報**とする）を「児童生徒本人と直接指導している教師のみ閲覧できる」ようにアクセス制御を施す方法を考える．今回，ABAC 方式に対応したアクセスポリシー記述言語仕様である XACML を用いて，ABAC 方式に基づくアクセス制御を実現する中継サーバを試作した．これを学校や学習塾などの共同指導が行われる教育機関の Web サイトに適用することで，ABAC 方式でのアクセス制御を実現する．

2. 関連研究と先行事例

本項では，XACML が用いられた研究や先行事例について述べる．

粒度の細かいアクセス制御のロールの扱いと高速化に関する研究

北川らは，医療電子カルテを例に，XACML を用いた粒度の細かいロールの扱いと，アクセス判定の高速化に関して考察している [4]．複数のロールをグループに分けることでポリシー記述を単純化し，アクセス頻度を考慮することでアクセス判定を高速化しようとしている．この研究では，XACML を用いてユーザが持つ複数のロールを一つずつ評価しているが，属性を評価基準としていない．本研究では属性を評価基準としているため，複数の属性を同時に評価できるような複雑なポリシーが必要になるが，ロールを一つずつ順番に評価する必要がなくなる．

スマートモバイルデバイスのビジネス活用を促進するデバイス保護管理技術

池田らは，セキュアな Android プラットフォームと連携して，Android の標準デバイス管理 API では制御できない OS やハードウェアに関連するイベントを，外部的・強制的に制御するアプリケーションを開発した [5]．この技術には，XACML をベースにした軽量なポリシー記述言語を利用した属性ベースアクセス制御モデルが採用されている．本研究はこの事例と同じく，生徒情報へのアクセスを外部的に制御することを目的としている．また，この事例ではタブレット上の限られたリソース下でのアクセス制御を想定しているが，本研究はリソースの潤沢なサーバ上でのアクセス制御を想定しているため，XACML をそのまま使用している．

XACML-Based Access Control for Decentralized Online Social Networks

Nasim らは，現在の中央集権型のオンラインソーシャルネットワーク (OSN) や Web サービスのプライバシー漏洩に対する脆弱性を指摘し，ユーザ一人ひとりが自分のデータを管理することを提案している [6]．この研究では，XACML を用いてユーザ同士の関係性（友人，家族など）を基にアクセス制御を行っており，生徒と教師の関係性を基にアクセス制御を行う本研究と同じ立場を採っている．

3. アクセス制御方式の比較

本項では，今日用いられているアクセス制御技術について示し，それぞれについて校務系情報・学習系情報のアクセス制御への適用可能性を検討する．

DAC（Discretionary Access Control）方式

DAC 方式とは，リソースの管理者がリソースに対してアクセス権限を設定する方式である．代表的な例として，Unix 系 OS などで実装されているファイルパーミッションが挙げられる．

DAC 方式はリソース本体に対してアクセス権限を直接設定するため，適切なアクセス権限を設定することができれば，アクセス制御の信頼性は高いと考えられる．しかし，リソース数が膨大になることが想定され

る場合，一つ一つのリソースに適切なアクセス権限を設定することは難しく，更新にかかるコストが大きい．

RBAC（Role-Based Access Control）方式

RBAC 方式とは，認可されていないユーザによるシステムへのアクセスを制限する手法である．ポリシーを柔軟に設定できるのが特徴である．

主な欠点として，企業の役職やポスト，所属部署など，ユーザに複数のロールを定義しなければならない場合，アクセス可否の判定の際にすべてのロールを一つずつ判定する必要が生じ，処理が膨大になることが挙げられる．また，アクセスする時間やユーザの位置情報などの情報を，アクセス可否の判定に利用できないなど，拡張性の低さも欠点となる．

校務系情報・学習系情報のアクセス制御を RBAC 方式で実装する場合，「教師」の区別を「1-1 担任の数学教師」「2-3 担任の英語教師」のようにそれぞれを一つのロールとして定義する方法と，教師，1-1 担任，数学，教師，2-3 担任，英語のように，1 ユーザに対し複数のロールを定義する方法がある．しかし，前者の場合はロールの細分化により管理コストが増大してしまい，後者の場合はアクセス判定の高速化の工夫が必要になる．

ABAC（Attribute-Based Access Control）方式

ABAC 方式とは，アクセスする主体の属性（名前や所属など），アクセス対象の属性（種類や所有者など），及びアクセスしている環境の属性（時間や場所など）に基づいて，特定のオブジェクトに対するアクセス可否を判定するアクセス制御モデルである．ABAC モデルを定義するアクセス制御ポリシー記述言語として，OASIS（Organization for the Advancement of Structured Information Standards）で標準化されている XACML が知られている．ユーザが持つ様々な属性をそれぞれ評価することができるため，柔軟なポリシー記述が可能である反面，RBAC 方式に比べてポリシー記述が複雑になる．

もっとも標準的な形式では，ABAC 方式は以下の 4 つの属性に基いてアクセス制御を行う．

Subject

アクセス主体を記述する属性（ユーザ ID，名前，ロールなど）

Action

オブジェクトに対するアクションを記述する属性（読み込み，削除，表示など）

Resource

アクセスされるオブジェクトを記述する属性（オブジェクトの種類（医療記録，銀行口座など），部門，場所など）

Environment

アクセスする時間やユーザの位置情報などの動的な要素を記述する属性

リソースそのものに直接アクセス権限を設定する DAC（Discretionary Access Control：任意アクセス制御方式）は，一つ一つの校務系情報・学習系情報に対してアクセス権限を設定するため非常に手間がかかり現実的ではない．RBAC 方式のアクセス制御では，例えば「教師」というロールに「校務系情報を閲覧する」権限を与えた場合に，すべての教師がすべての校務系情報を閲覧する権限を持つことになる．ABAC 方式を活用すれば，指導クラスや指導科目を「属性」として定義しアクセスポリシーを定めることができ，柔軟なアクセス制御の実現が期待できる．

4. ポリシー記述言語 XACML

4.1 XACML 規格

XACML（eXtensible Access Control Markup Language）とは，ABAC 方式を実現するアクセス制御技術として，産業標準化団体である OASIS で標準化されているポリシー記述言語である [7]．最新版の XACML 3.0 は 2013 年に策定された．

XACML は，ポリシー決定ポイント（PDP）とポリシー執行ポイント（PEP）が，ユーザの属性情報から形成された XACML 承認リクエストを基にアクセスを認可する．XACML のポリシーは，「アクセス制御の対象」，「ルール結合アルゴリズム識別子」，「ルールの集合」，「責務（obligation）」から成り立つ．（図 1）

アクセス制御の対象は正規表現で表現することができ，例えば Web サイトのある階層すべてに同一のアクセス制御を施す場合，一つのルールで記述することができる．また，obligation は条件付きの認可を定義するためのオプションで，通常は認可／否認の決定とは別にアクションを定義するものであるが，本研究では obligation が満たされるかどうかを認可の判断に含めるように設計した．教師から「児童生徒の成績」へのアクセス要求が行われた際，「直接指導を行っていること」を obligation として定義することで，無関係の教師からのアクセス要求を否認することができる．

XACML は非常に強固なセキュリティを提供するが，一般的なデータに対しては過剰であると考えられていた．これまでは主にボーイング，銀行，保険会社など機密性の高い情報を扱う組織で利用されていたが，XACML を利用した IoT プラットフォームの「FIWARE[*1]」に注目が集まっており，電子情報の比重が増加している昨今，XACML の需要は大きくなると予想される．

[*1] https://www.fiware.org

```
<Policy>
<RuleCombiningAlg = xxx>     //ルール結合アルゴリズム識別子
  <Target>xxx.com/xxx/.*<\Target>     //アクセス制御の対象
  <Rule>Rule1</Rule>
  <Rule>Rule2</Rule>
  <Obligation     //obligation（オプション）
    send e-mail to administrator
  </Obligation>
<\Policy>
```

図 1 XACML ポリシー記述例
Fig. 1 Example of XACML policy

4.2 XACML 処理系 AuthZForce

AuthZForce は，「FIWARE」が開発したオープンソースの ABAC フレームワークである[*2]．本研究では，アクセス制御のための PDP として利用した．

5. 共同指導体制におけるアクセス制御要求と解決策

本項では，共同指導体制におけるアクセス制御要求と解決策を示す．なお要求については，「教育情報セキュリティポリシーに関するガイドライン」[3] の校務系情報・学習系情報と教職員の業務負担軽減の項を基にした．また，下記の情報公開範囲は本研究で想定した一例である．

校務系情報のアクセス制御に対する要求

校務系情報について，クラス担任の教師は自分のクラスの児童生徒のすべての情報を閲覧でき，授業を受け持つ教師は指導科目の情報のみを閲覧でき，その他の教師は一切の情報を閲覧できないことが求められる．

学習系情報のアクセス制御に対する要求

学習系情報について，児童生徒は自分の情報のみを閲覧できるように，教師については校務系情報と同様にアクセスを制御することが求められる．

アクセス制御に対する解決策

指導関係を評価するため，「所属クラス」「指導科目」「指導クラス」という属性をデータベースに定義する．教師からのアクセス要求に対し，指導関係にある児童生徒の情報へのアクセス要求かどうかを obligation を用いて判別する．例として，ある教師 A に自分が担任するクラスの生徒 A の成績情報の閲覧権限を与えるルールを図 2 に示す．なお，この例では XACML3.0 ではなく記法が簡易な XACML2.0 のタグを用いた．児童生徒と教職員の属性情報をデータベースで管理し，それを基に XACML 承認リクエストを形成すれば，データベース内の属性情報を書き換えるだけで情報の公開範囲を変更できる．

6. ポリシー記述言語対応中継サーバの仕様

本項では，開発したポリシー記述言語対応中継サーバの仕様について示す．

[*2] https://authzforce.ow2.org/

```
<Rule Effect = Permit>
  <Subjects>教師<\Subjects>
  <Resources>生徒 A<\Resources>
  <Actions>Read<\Actions>
  <Obligation>
    「生徒 A の所属クラス」==「教師 A の所属クラス」
  <\Obligation>
<\Rule>
```

図 2 生徒 A の成績情報に関するルール記述例
Fig. 2 Example rule for student A's record

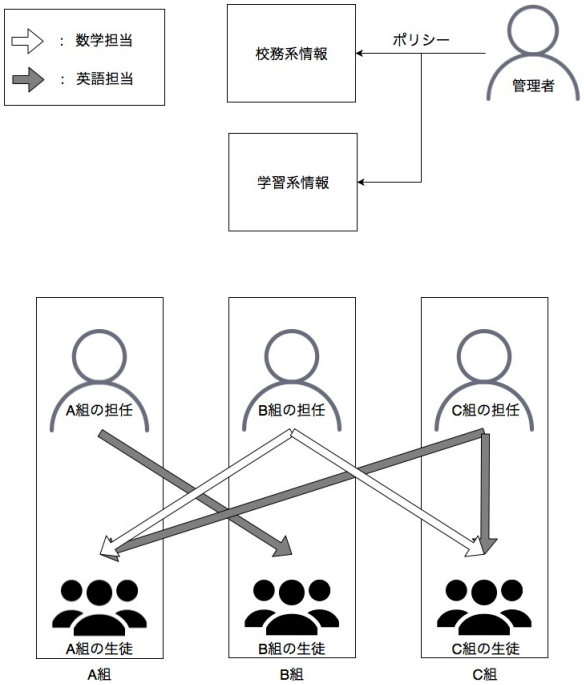

図 3 学校の組織構造の例
Fig. 3 The organizational structure in example school

6.1 開発の目的

教師が直接指導している児童生徒の情報のみを閲覧できるようにするために，データベースに格納されたユーザの属性を利用した ABAC 方式のアクセス制御を実現するポリシー記述言語対応中継サーバ（以下，ABAC サーバ）を開発する．具体例として，校務系情報を Web ブラウザを通して閲覧できるシステムを導入した学校での利用を想定する．

6.2 開発の基本方針

対象となるユーザ

図 3 のように，1 クラスに 1 人ずつ担任教師が存在し，教師は 1 つまたは複数のクラスで授業を受け持つような組織構造をもつ学校に所属する児童生徒と教師と管理者

設計方針

児童生徒と教師がもつ名前，ロール，所属クラスなど

の属性をもとにアクセス要求を評価し，認可されたユーザに対してのみアクセスを許可する．アクセス制御には ABAC サーバを用いる．ユーザは初めに自分の ID とパスワードを用いてシステムにログインすれば，一般的な Web ページ閲覧と同じようにシステムを利用できる．ユーザのアクセス要求が否認された場合は 403 を返す．

実装方針

アクセス制御には，ABAC 方式を採用したオープンな処理系をもつポリシー記述言語である XACML を採用する．アクセス要求を評価する PDP には AuthZForce を使用する．アクセス制御を施す情報リソースは，校務系情報を閲覧できる Web サイトとする．アクセス制御を行う PEP は Python を用いて実装する．PDP に送信する XACML 承認リクエストは，中継サーバ内のテンプレートファイルを書き換えることで生成する．

6.3 管理機能

ABAC サーバがもつ管理者用機能について以下に示す．

- ユーザの属性情報はデータベースで管理し，管理者はデータを入力・変更・削除できる
- データベース内の属性情報をアップデートするだけでアクセス権限を変更できる
- アクセス制御はポリシー記述言語 XACML を用いて行い，管理者はポリシーを追加・変更・削除できる

7. ポリシー記述言語対応中継サーバの実装

本節では，開発したポリシー記述言語対応中継サーバの実装について述べる．

7.1 ポリシー記述言語対応中継サーバの概要・機能

ABAC サーバは，教師のもつ属性（担任クラスや指導科目）に応じて，校務系情報・学習系情報の閲覧制限を実現するためのポリシー記述言語対応中継サーバである．ABAC サーバは Web サイト閲覧においてページ単位でのアクセス制御を行うことができる．

ABAC サーバは以下の機能をもつ．

- 教師・児童生徒の認証機能
- ユーザの Web サイトへのアクセス要求を XACML 承認リクエストに変換する機能
- PDP からのレスポンスに Obligation が含まれていた場合，Obligation によって指定された条件が満たされるかどうかを判断し，アクセス可否を決定する機能
- アクセス要求が認可された際，ユーザのアクセス要求を Web サイトに中継する機能
- アクセス要求が否認された際，ユーザのアクセス要求を Web サイト上のエラーベージへのアクセス要求に変換する機能

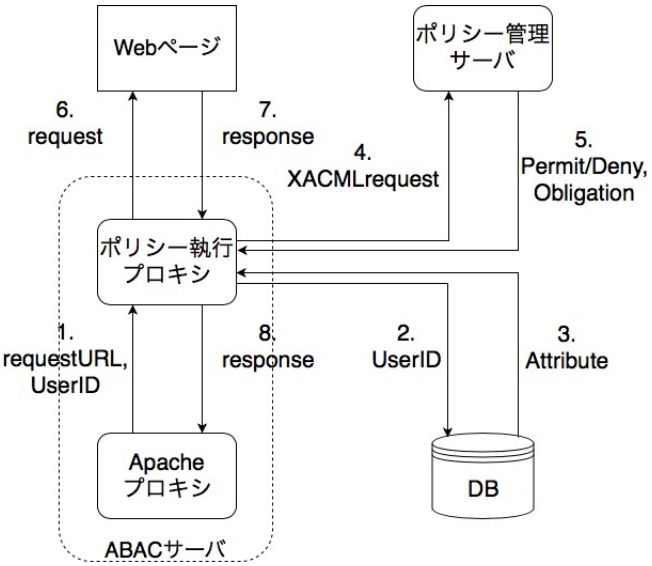

図 4 試作システムの全体構成
Fig. 4 Dataflow diagram of the developed ABAC server

7.2 試作システムの全体構成

試作システムの全体構成を図 4 に示す．ABAC サーバは，BASIC 認証を行う Apache プロキシ，PEP の役割を果たすプロキシサーバー（以下，ポリシー執行プロキシ）の 2 つのモジュールで構成されている．ABAC サーバは，PDP の役割を果たす AuthZForce（以下，ポリシー管理サーバ）とユーザの属性を登録するフラットファイルデータベース（以下，データベース），アクセス制御を施す Web サイトと連携して機能する．これらをまとめて試作システムとする．

7.3 ポリシー執行プロキシ

ポリシー執行プロキシは，Python で書かれたオープンソースのプロキシサーバプログラムである”PyProxy”[*3]に PEP の機能を追加実装したものである．ポリシー執行プロキシは，Apache プロキシから受信したリクエスト文から，ユーザがアクセスしようとしているリソースのパス（以下，要求リソースパス）と base64 エンコードされたユーザの ID（以下，ユーザ ID）を抽出する．次に，ユーザ ID を用いてデータベースからユーザの属性を取り出す．そして，ポリシー執行プロキシ内の XACML 承認リクエストテンプレートにユーザの属性と要求リソースパスを書き込む．こうして生成した XACML 承認リクエストを，REST API を利用してポリシー管理サーバに転送し，ポリシー管理サーバからのレスポンスの内容に応じて以下の処理を行う．

- Permit の場合，要求リソースへの GET リクエストを行う
- Deny の場合，Web サイトにエラーページの GET リクエストを行う

[*3] https://github.com/magayengineer/PyProxy

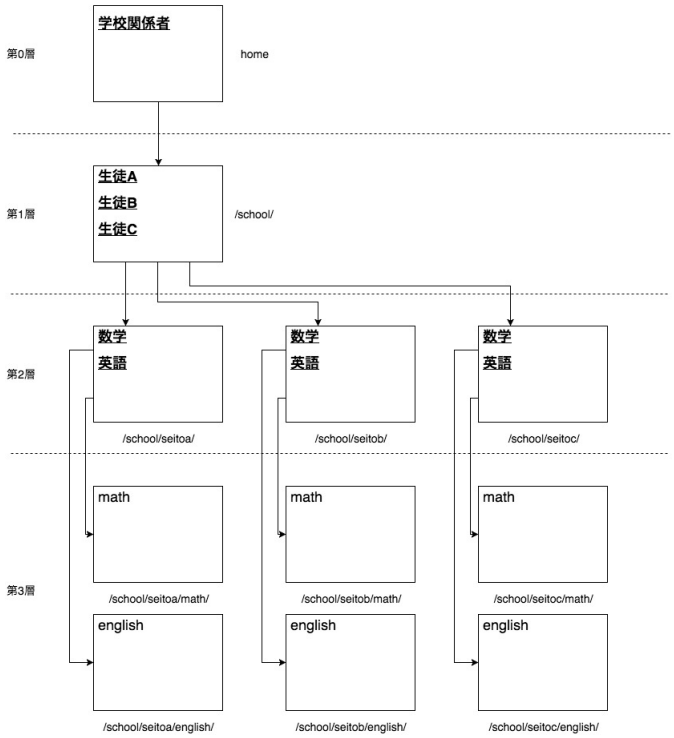

図 5 Web サイトの構成
Fig. 5 The website structure of an imaginary school

```
<PolicySet
  <Policy PolicyId="school">
  #schoolディレクトリに関するポリシー
    <Resource>/school/</Resource>
    <Rule Effect="Permit">
    #メールアドレスのドメイン名がexample.ed.jpの場合readできる
      <Action>read</Action>
      <Condition>
        <Function "rfc822Name-match"/>
          <AttributeValue>example.ed.jp</AttributeValue>
      </Condition>
    </Rule>
  </Policy>
  <Policy PolicyId="subject">
  #subjectディレクトリに関するポリシー
    <Resource>
      <Match "string-regexp-match">/school/.*/.*/.*</Match>
    </Resource>
    <Rule Effect="Permit">
    #teacherロールを持つアクセス要求に対し，"check-sub"のObligationが発行される
      <Action>read</Action>
      <Condition>
        <Function "string-equal"/>
          <AttributeValue>teacher</AttributeValue>
      </Condition>
      <ObligationExpression ObligationId="check-sub" FulfillOn="Permit"/>
    </Rule>
  </Policy>
  <Policy PolicyId="student">
  #studentディレクトリに関するポリシー
    <Resource>
      <Match "string-regexp-match">/school/.*/.*</Match>
    </Resource>
    <Rule Effect="Permit">
    #studentロールを持つアクセス要求に対し，"check-name"というObligationが発行される
      <Action>read</Action>
      <Condition>
        <Function "string-equal"/>
          <AttributeValue>student</AttributeValue>
      </Condition>
      <ObligationExpression ObligationId="check-name" FulfillOn="Permit"/>
    </Rule>
    <Rule Effect="Permit">
    #teacherロールを持つアクセス要求に対し，"check-class"というObligationが発行される
      <Action>read</Action>
      <Condition>
        <Function "string-equal"/>
          <AttributeValue>teacher</AttributeValue>
      </Condition>
      <ObligationExpression ObligationId="check-class" FulfillOn="Permit"/>
    </Rule>
  </Policy>
</PolicySet>
```

図 6 記述したポリシーの例
Fig. 6 Example access policy

ポリシー管理サーバからのレスポンスに ObligationID が含まれている場合，ポリシー執行プロキシは ObligationID の値に応じてユーザ属性の評価を行い，Permit/Deny を決定する．ObligationID とその内容を以下に示す．

check-name
自分の情報へのアクセス要求かどうかを判定する．

check-class
担任または指導するクラスの生徒の情報へのアクセス要求かどうかを判定する．

check-sub
担任クラスの生徒の情報へのアクセス要求かどうか，または指導クラスの生徒の指導科目の情報へのアクセス要求かどうかを判定する．

Web サイトからのレスポンスは，Apache プロキシを通してアクセス元の HTTP クライアントに送られる．

7.4 ポリシーの記述例

試作システムを用いたアクセス制御を施す Web サイトの構造を図 5 に示す．四角形は Web ページを表している．下線のついた文字列はリンクを表し，矢印は各リンクの参照を表す．四角形の右・下に書かれた文字列は URL のパスであり，home はホームページを表す．

この Web サイトのアクセス権限について記述したポリシーの例を図 6 に示す．

school ポリシーは，図 5 における第 1 層へのアクセスを評価するポリシーである．学校関係者だけがもつメールアドレスのドメイン名を評価基準にしている．subject ポリシーは，図 5 における第 3 層以降へのアクセスを評価するポリシーである．教師のアクセスに対して”check-sub”の Obligation を PEP に返す．student ポリシーは，図 5 における第 2 層へのアクセスを評価するポリシーである．生徒のアクセスに対して”check-name”の Obligation を，教師のアクセスに対して”check-class”の Obligation を PEP に返す．

XACML 規格では，ポリシーは上から順に評価されるので，生徒が第 3 層以降へアクセスする際にも student ポリシーでアクセスを評価することができる．

各階層ごとにリソースパスの正規表現でポリシーを定義しているため，生徒が増えた場合も層の構造が変わらなければポリシーセットに変更を加える必要はない．

8. 動作検証

本節では，動作検証にあたって設定した属性情報や，アクセス制御の検証の流れを示す．

8.1 前提条件

3 人の生徒の数学と英語の学習系情報を例として，3 人の生徒と 3 人の教師に対するアクセス制御の動作検証を行う．データベースの値を変更することでアクセス権限が正しく変更されるかどうかを検証するため，2 種類の属性データベースに対してそれぞれ検証を行う．ユーザに与え

表 1 ユーザの属性 A
Table 1 Sample users and their attributes A

ユーザ	メールアドレス	名前	ロール	指導科目	所属クラス	指導クラス
生徒 A	seitoa@example.ed.jp	seitoa	生徒	なし	1-1	なし
生徒 B	seitob@example.ed.jp	seitob	生徒	なし	1-2	なし
生徒 C	seitoc@example.ed.jp	seitoc	生徒	なし	1-3	なし
教師 A	kyoushia@example.ed.jp	kyoushia	教師	数学	1-1	1-1,1-3
教師 B	kyoushib@example.ed.jp	kyoushib	教師	数学	1-2	1-2
教師 C	kyoushic@example.ed.jp	kyoushic	教師	英語	1-3	1-1,1-2

表 2 ユーザの属性 B
Table 2 Sample users and their attributes B

ユーザ	メールアドレス	名前	ロール	指導科目	所属クラス	指導クラス
生徒 A	seitoa@example.ed.jp	seitoa	生徒	なし	2-3	なし
生徒 B	seitob@example.ed.jp	seitob	生徒	なし	2-1	なし
生徒 C	seitoc@example.ed.jp	seitoc	生徒	なし	2-4	なし
教師 A	kyoushia@example.ed.jp	kyoushia	教師	数学	2-2	2-1,1-4
教師 B	kyoushib@example.ed.jp	kyoushib	教師	数学	2-4	2-3
教師 C	kyoushic@example.ed.jp	kyoushic	教師	英語	1-5	1-1,1-2

表 3 実行環境
Table 3 Test Environment

項目	内容
機種	MacBook Pro (13-inch, 2016)
CPU	2.9GHz Intel Core i5
メモリ	16 GB 2133 MHz LPDDR3
Apache のバージョン	2.4.28
PyProxy のコミット番号	228b84e
AuthZForce のバージョン	5.4.1
Docker のバージョン	17.12.0-ce-mac49

られる属性を表 1，表 2 に示す．以下，表 1 の属性に対する検証をパターン A，表 2 の属性に対する検証をパターン B とする．

8.2 実行環境

動作検証を実行したマシン環境と使用したソフトウェアのバージョンを表 3 に示す．今回の動作検証では，図 4 に示す Web サイトとして Xdomain レンタルサーバ上に構築した Web サイトを用い，ABAC サーバ部は検証マシン上の仮想サーバとして動作させた．また，ポリシー管理サーバの AuthZForce は配布されている Docker イメージを検証マシン上の Docker 環境で動作させている．検証マシン上の Web ブラウザのプロキシに ABAC サーバの Apache プロキシを設定することで，ABAC サーバを用いた既存 Web サイトへのアクセス制御動作を検証することができる．なお表 1，表 2 のユーザアカウントは Apache プロキシ上の.htpasswd ファイルに登録している．

8.3 実験用データベースの内容

表 1 のユーザ属性を表すデータベースの内容を表 4 に，表 2 のユーザ属性を表すデータベースの内容を表 5 に示す．なお，データベースは第一正規形で表現する．

表 4 データベースの内容 A
Table 4 Database A

メールアドレス	名前	ロール	指導科目	所属クラス	指導クラス
seitoa@example.ed.jp	seitoa	student	NULL	1-1	NULL
seitob@example.ed.jp	seitob	student	NULL	1-2	NULL
seitoc@example.ed.jp	seitoc	student	NULL	1-3	NULL
kyoushia@example.ed.jp	kyoushia	teacher	math	1-1	1-1
kyoushia@example.ed.jp	kyoushia	teacher	math	1-1	1-3
kyoushib@example.ed.jp	kyoushib	teacher	math	1-2	1-2
kyoushic@example.ed.jp	kyoushic	teacher	english	1-3	1-1
kyoushic@example.ed.jp	kyoushic	teacher	english	1-3	1-2

表 5 データベースの内容 B
Table 5 Database B

メールアドレス	名前	ロール	指導科目	所属クラス	指導クラス
seitoa@example.ed.jp	seitoa	student	NULL	2-3	NULL
seitob@example.ed.jp	seitob	student	NULL	2-1	NULL
seitoc@example.ed.jp	seitoc	student	NULL	2-4	NULL
kyoushia@example.ed.jp	kyoushia	teacher	math	2-2	2-1
kyoushia@example.ed.jp	kyoushia	teacher	math	2-2	1-4
kyoushib@example.ed.jp	kyoushib	teacher	math	2-4	2-3
kyoushic@example.ed.jp	kyoushic	teacher	english	1-5	1-1
kyoushic@example.ed.jp	kyoushic	teacher	english	1-5	1-2

表 6 アクセスの検証結果 A
Table 6 Access result A

ユーザ	home	/school/	/school/seitoa/			/school/seitob/			/school/seitoc/		
			root	math	english	root	math	english	root	math	english
生徒 A	○	○	○	○	○	×	×	×	×	×	×
生徒 B	○	○	×	×	×	○	○	○	×	×	×
生徒 C	○	○	×	×	×	×	×	×	○	○	○
教師 A	○	○	○	○	○	×	×	×	○	○	×
教師 B	○	○	×	×	×	○	○	○	×	×	×
教師 C	○	○	○	×	○	○	×	○	○	○	○

表 7 アクセスの検証結果 B
Table 7 Access result B

ユーザ	home	/school/	/school/seitoa/			/school/seitob/			/school/seitoc/		
			root	math	english	root	math	english	root	math	english
生徒 A	○	○	○	○	○	×	×	×	×	×	×
生徒 B	○	○	×	×	×	○	○	○	×	×	×
生徒 C	○	○	×	×	×	×	×	×	○	○	○
教師 A	○	○	×	×	×	○	○	×	×	×	×
教師 B	○	○	○	○	×	×	×	×	○	○	○
教師 C	○	○	×	×	×	×	×	×	×	×	×

8.4 検証結果

各ユーザ ID においてアクセスが認可されたかどうかについて，パターン A を表 6 に，パターン B を表 7 に示す．認可されたアクセスを○，否認されたアクセスを×で表している．パターン A とパターン B で，生徒 A の情報を例に 3 人の教師のアクセスがどのように処理されるかを確かめる．

パターン A では，教師 A は生徒 A のクラスの担任であるためすべての情報にアクセスでき，教師 B は生徒 A を直接指導していないためアクセスができず，教師 C は生徒 A のクラスで英語を教えているため英語の情報にアクセスできる．パターン B では，教師 A と教師 C は生徒 A を直接指導していないためアクセスができず，教師 B は生徒 A のクラスで数学を教えているため数学の情報にアクセスできる．

以上のことから，意図したとおりにアクセス制御が機能していることが確認できる．

9. 考察と今後の課題

本節では，中継サーバ適用方法についての考察と今後の課題について述べる．

9.1 ポリシー記述言語対応中継サーバに対しての考察

本研究では，想定した学校の構造について，クラス単位で行う授業のみを定義していた．しかし，生徒が自由に選択する授業や補講授業などについても，生徒の「所属クラス」に別のクラス属性を追加することでアクセス制御を施すことができると考えられる．例えば，地理を選択した生徒のグループを geoA，数学の補講を受ける生徒のグループを math-hokou などと定義することが考えられる．

また，本報告では 3 人の生徒と 3 人の教師の属性情報をデータベースに登録して動作検証を行ったが，実際の教育現場ではこれよりも多くのユーザの属性情報を登録する必要がある．データベースのサイズの増大が ABAC サーバの動作に与える影響を検証する必要がある．

9.2 複数の学校による運用

本研究では，情報リソースが単一の学校で運用されることを前提に実装・動作検証を行ったが，「教育情報セキュリティポリシーに関するガイドライン」では，複数の学校が 1 つの文書サーバ，データベースを共有する形を採っている．今後の課題として，複数の学校による運用を想定した実装・動作検証が挙げられる．

10. 結言

本研究では，ポリシー記述言語を用いて属性情報によるアクセス制御を実現する中継サーバを開発した．Web サイトのリソースに対して個別にアクセス権限を設定するのではなく．データベースにユーザの属性を登録するだけで適切なアクセス権限を設定するできることを確認した．

謝辞 本研究は JSPS 科研費 18K11568 の助成を受けたものです．

参考文献

[1] 官邸：日本再興戦略 2016，官邸（オンライン），入手先 ⟨https://www.kantei.go.jp/jp/singi/keizaisaisei/pdf/2016_zentaihombun.pdf⟩（参照 2018-05-17）．
[2] 神谷加代, 高木大地, 中野信二：教育 ICT ガイドブック, 総務省 情報流通行政局情報通信利用促進課（オンライン), 入手先 ⟨http://www.soumu.go.jp/main_content/000492552.pdf⟩ (参照 2018-05-17)．
[3] 文部科学省：教育情報セキュリティポリシーに関するガイドライン，文部科学省（オンライン），入手先 ⟨http://www.mext.go.jp/a_menu/shotou/zyouhou/detail/_icsFiles/afieldfile/2017/10/18/1397369.pdf⟩（参照 2018-05-17）．
[4] 北川直毅，吉川正俊：粒度が細かいアクセス制御のロールの扱いと高速化に関する研究，*DEWS2006*, Vol. 6B-i8, pp. 1–8（オンライン），入手先 ⟨http://www.ieice.org/˜de/DEWS/DEWS2006/doc/6B-i8.pdf⟩ (2006)．
[5] 池田竜朗，森尻智昭，阿部真吾：スマートモバイルデバイスのビジネス活用を促進するデバイス保護管理技術，東芝（オンライン），入手先 ⟨https://www.toshiba.co.jp/tech/review/2014/01/69_01pdf/a07.pdf⟩（参照 2018-05-17）．
[6] Nasim, R. and Buchegger, S.: XACML-Based Access Control for Decentralized Online Social Networks, *Proceedings of the 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing*, UCC '14, Washington, DC, USA, IEEE Computer Society, pp. 671–676 (online), DOI: 10.1109/UCC.2014.108 (2014)．
[7] Lockhart, H., Parducci, B. and Levinson, R.: OASIS eXtensible Access Control Markup Language (XACML) TC, OASIS (online), available from ⟨https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml⟩ (accessed 2018-05-22)．