社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2006－DBS－140（Ⅰ）（24）
2006／7／12

XMLデータベースにおける個人情報のアクセスコントロール方式の設計

福島 卓也 † 遠山 元道 ‡

† 慶應義塾大学大学院 理工学研究科 開放環境科学専攻
‡ 慶應義塾大学 理工学部 情報工学科
E-mail: † takuya@db.ics.keio.ac.jp, ‡ toyama@ics.keio.ac.jp

近年個人情報保護法の施行などにともない，個人情報管理の関心が高まっているといえる．個人情報はその情報の持ち主が「どの情報を誰と共有するか」というポリシーが人によって異なるという性質がある．したがって個人情報をデータベースで管理するときも各人のポリシーに応じた柔軟なアクセス制御が要求される．本稿ではデータベース中のデータに対して所有者を定義し，データの所有者とデータにアクセスするユーザーとの間に定義されたルールに基づいて制御を行う Rule-Based Information Masking を提案し，個人情報を含む XML 文書に対してこれを適用する．

キーワード：XML，　アクセス制御，　個人情報

Access Control Model for Personal Data in XML Database

Takuya FUKUSHIMA † Motomichi TOYAMA ‡

†School of Science for OPEN and Environmental Systems,
Faculty of Science and Technology, Keio University.
‡Department of Information and Computer Science, Faculty of Science and Technology,
Keio University.
E-mail : †takuya@db.ics.keio.ac.jp ‡toyama@ics.keio.ac.jp

With the enforcement of personal information protection law, concern over management of personal data is growing nowaday. Personal data have a feature that the access policies which show owner is sharing ”which information with whom” differ among people. In this paper, we propose new access control model called Rule-Based Information Masking. RBIM defines owners of data in database, and it computes access rights by means of rules defined between owner of data and user accessing to the data. We applied this model to XML document which includes personal data.

keyword : XML , Access Control , Personal Information

1　はじめに

近年個人情報への関心が高まっており，個人情報の管理に関する世間の関心が高まっている．個人情報の管理において懸念されることは情報の漏洩であり，自分の情報が意図しない第三者に閲覧，使用されないことが求められている．

まったく同じ情報であってもその情報に対する考え方は人によって様々である．例えば同じ部署に勤める同僚にならば「住所や電話番号を知られてもよい」と考える人もいれば,「電話番号は知られてもいいが住所は知られたくない」と考える人もいる．さらに「同僚にはクレジットカードの番号を知られたくない」と考える人でも,「配偶者にはクレジットカードの番号を知られてもよい」と考えるといったように，同じ属性の情報でも個人間の関係によって情報への考え方が異なる点が個人情報の特徴であるといえる．

以上から個人情報は情報の持ち主が「どの情報を誰と共有するか」という点を考慮してデータベースで管理する必要があると考えられる．しかし，今日存在するデータベースにおいてデータの持ち主に関しては考えられておらず，その持ち主のポリシーに基づいたアクセス制御を行うことは非常に困難である．

また，近年ネイティブ XML データベースなど，XML データに対してユーザーが直接問い合わせを行い，情報を取得する機会が増大している．XML に対するセキュリティに関する話題ではOASIS によって，アクセス制御のフレームワークとしてXACML[1] が策定された．しかし XACML で規定されているアクセス制御は XML 問い合わせに対するアクセス制御として十分なものとはいえない．XML 問い合わせに対するアクセス制御は，他の研究や今日存在する商用ネイティブ XML データベースでもサポートしているが，先に述べたようなデータの持ち主を考慮したものはなく個人情報に対して十分な制御をできるものはない．

そこで本稿では個人間の関係に基づいたアクセス制御を行う Rule-Based Information Masking (RBIM) を提案し，個人情報を含む XML 文書に対する柔軟なアクセス制御を実現する．RBIM ではデータベース中の特定の範囲にデータの所有者を定義し，そのデータにアクセスしようとする人物とアクセスされるデータの所有者との間に定義されたアクセスルールに基づいてアクセス制御を行う．

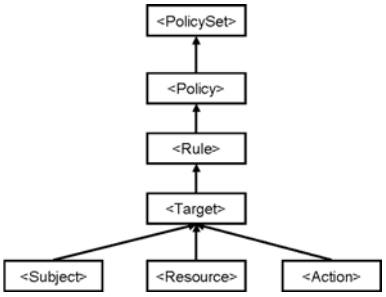

図 1: XACML のポリシー記述言語の構造

本稿の構成は以下の通りである．2 章では XML のアクセス制御およびアクセス制御モデルなどの関連研究を紹介し，個人情報管理におけるそれらの手法の問題点について述べる．3 章では RBIM の概念について述べ，4 章では RBIM の概念を XML に拡張し，5 章では検討事項および今後の課題について述べる．最後に 6 章にまとめを述べる．

2　関連研究と問題点

本章ではアクセス制御に関する関連研究について述べる．まず 2.1 節では，XML のアクセス制御の標準的なフレームワークである XACML や，XML 問い合わせに対する研究事例を紹介する．次に 2.2 節では，今日のアクセス制御モデルとして一般的な Role-Based Access Control (RBAC) を紹介し，個人情報管理の観点から見た RBAC の問題点について議論する．

2.1　XML のアクセス制御

XML のアクセス制御のフレームワークとして OASIS が標準化している XACML[1] があげられる．XACML のポリシー記述言語の構造を図 1 に示す．アクセス規則は XML 形式で記述されたルール (Rule) からなり，ルールでは主体 (Subject)，資源 (Resource)，動作 (Action) からなる対象 (Target) に対する規則を定めることができる．動作は読み込み，書き込み，削除などの動作のほかに任意の動作を定義することが可能である．また，複数のルール

を結合してポリシー (Policy) を定めたり，複数のポリシーからなるポリシー集合 (PolicySet) を作成することも可能である．

このようにXACMLは非常に複雑なアクセスルールを記述することが可能であるが，元々がWebサービスのセキュリティフレームワークとして標準化された仕様であるため，XML問い合わせに対するセキュリティモデルとして十分ではないといえる．例えばXACMLではルールを適用する資源 (Resource) を指定する際，資源識別子を用いたXMLデータの外部実体レベルの指定しかできず，内部実体レベルで文書の内部構造に即して制御の対象範囲を指定するのは困難である．

XACMLの他にもXMLのアクセス制御に関する研究は数多くなされている．Damianiら [2] の手法では，あるユーザーグループに属しているユーザーが文書にアクセスする際，そのユーザーグループに設定されたルールに基づいて，アクセスが許可された部分からなるビューを計算することで，許可していない部分をユーザーにアクセスされることを防いでいる．しかしこの手法ではアクセスの要求がある度にビューを構築しないといけないため，ビュー構築のコストが非常に大きくなる点が問題であるといえる．これに対してFanら [3] の手法では，元のドキュメントに対するクエリの結果がビューに対するクエリの結果と等価になるように，クエリの書き換えを行う．これにより実際にビューを生成することなくビューを生成するときと同じ結果を得ることを可能にしており，計算量を大幅に削減している．

しかし，いずれの手法もデータがだれの所有物かという点については議論されておらず，それぞれのデータの所有者とデータにアクセスしようとしているユーザーの個人間の関係に基づいた制御を行うことは不可能である．そのため，個人情報に関するアクセス制御としては上記の手法は不十分であるといえる．

2.2 Role-Based Access Control

本節では今日アクセス制御方式として普及しているRole-Based Access Control (RBAC) について述べ，個人情報管理におけるRBACの問題点について考察する．RBACでは「データアクセスの権限は個々のユーザーにではなく，仕事の役割 (ロール) に対して割り当てられるべきである」という概念に基づいて制御が行われる．ロールに対して権限が割り当てられ，ユーザーとロールが多対多で関連付けられる事により，ユーザーに複数のロールを割り当てることを可能にし，複雑なアクセス制御を表現している．また図2の (a) に示すように．ロールを半順序関係で示される階層構造を持つことが可能で，下位ロールの権限を継承することができる．

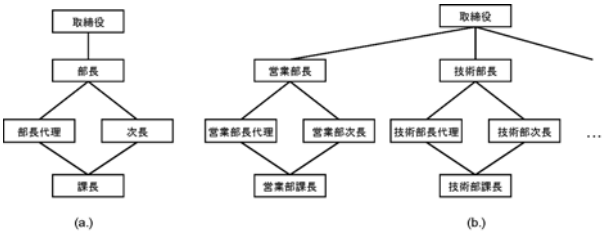

図 2: 階層型ロール

このようにRBACは組織の権限を管理するのに非常に適した手法であり，商用RDBをはじめとした多くのデータベースでも採用されている．しかし，RBACには以下のような問題点が存在する．

1. ビューによる管理の困難
2. データベースと分離したロール管理
3. アクセス権管理の集中

一つ目の問題はRBACではビューに基づいたアクセス制御を行うため，ロールごとにビュー定義をしなければいけないという点である．簡単のため関係データベースで示した例を図3に示す．基底表 R に関してユーザーAのロールにアクセスを許可された部分からなるビューを V_A，ユーザーBのロールに対するものを V_B とする．このときユーザーはそれぞれ自分にアクセス権限があるビューに対して質問文を生成する必要があり，自分がどのビューにアクセスを許可されているのかを意識して質問文を発行しないといけない．そのため発行する質問文もユーザーによって異なる．さらに，複雑なロール管理がこの事態を悪化させている．例えば図2(a) のロールが定義されているときに，「社員は自分と同じ部署にいる社員データにしかアクセスできない」という新しいポリシーを追加するとき，図2(b) のように部署ごとにロールを作らなければならない．

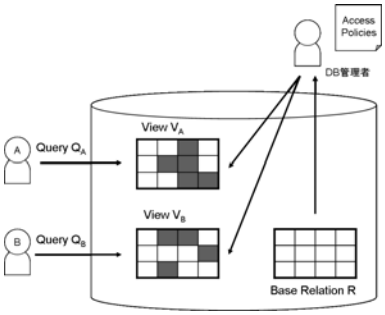

図 3: ビューによるアクセス制御

このように新しく生成されるロールの分だけビューを生成しないといけないため，データベース管理者の負担の観点で見ても問題であるといえる．

第二にロールなどのアクセスポリシーの管理はデータベースのシステムとは独立してなされることも管理者の負担を助長しているといえる．たとえばある社員が技術部から営業部に異動する場合，データベース管理者はデータベースの内容を変更するだけでなく，その社員に関するロールも変更するといったように，アクセスポリシーも変更する必要がある．また，一点目に述べたビューの生成に関する弊害はアクセスポリシーの変更がデータベースに反映されない例であるといえる．これらの変更作業をデータベース管理者は手作業で行わなければならない．

第三に RBAC ではアクセスポリシーの付与はロール単位で行うため，同じロールに属するユーザー間では同じアクセスポリシーが採用される．そのためデータの所有者によって個人情報に関する取り扱いポリシーは異なるにも関わらず，データの所有者個々のポリシーに基づいたアクセス制御をすることは不可能である．さらに現状では一部の管理者権限をもつユーザーしかアクセス権限の付与ができず，個人情報の所有者が自分のデータに対してアクセス権限の付与を行うことができないため，すべてのアクセス権限の付与はデータベース管理者の負担となる．

3 Rule-Based Information Masking

本章では先に述べた RBAC などのビューを用いた方式に替わるアクセスコントロール方式として Rule-Based Information Masking (RBIM) を提案する．RBIM は従来のアプローチと比較して以下の点で優れているといえる．

1. ビューを必要としないアクセス管理
2. データベースと連動したアクセス管理
3. アクセス管理権限の分散，柔軟化

本章では関係データベースにおけるモデルについて考え，XML におけるモデルは次章詳しく述べることとする．

3.1 RBIM データベースの概観

図 4 に RBIM データベースの概念図を示す．本手法ではユーザーがクエリを発行してアクセスしようとした基底表に対して，そのユーザーに定められたアクセスルールとデータベース中に蓄えられた情報からユーザーに許可されていない部分を隠したビューを導出する．そしてこの導出されたビューに対してクエリを実行することで，閲覧が許可されていない情報が隠された結果をユーザーに返すことができる．なおアクセスが許可されていない箇所に関しては新しい null 値の一種として denied という値に置き換えてユーザーに結果を返すものとする．

本手法ではユーザーがシステムに質問文を発行した時点でビューを導出するため，ユーザーは基底表に対して質問文を発行すれば自動的にアクセス制御がされた結果を得ることができる．これによって自分が許可されているビューの存在を意識することなくユーザーごとのアクセス制御を可能にする．

また，RBIM ではアクセスポリシーをもとにビューを導出するのでアクセスポリシーやデータベースの内容の変更が生成されるビューに即座に反映される．これは RBAC などのビューを定義する手法と大きく異なる点であり，提案手法では管理者の負担を大幅に削減できる．

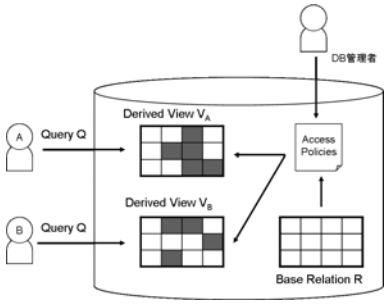

図 4: RBIM データベース

3.2 提案モデル

関係データベースにおける RBIM のモデルを以下のように定義する．

定義 1 データベース中の全ての関係 R_i は主キー $PK(R_i)$ を持つ．クエリを発行するユーザーのユーザー ID はシステムに登録されており U と表す．

ここで簡単のために，主キー $PK(R_i)$ は単一の属性からなるものとする．

定義 2 $\alpha(s, o, r.a)$ はアクセス判定関数というブーリアン関数とする．ここで s はアクセス主体，o はアクセス対象，$r.a$ は関係 r 中の属性 a を表す．アクセス判定関数はあるタプル $t \in r$ における属性 a について，ユーザー s がアクセス可能のとき true となる．ここで t は属性 r における主キー $o \in PK(r)$ によって一意に識別されるものとする．

定義 3 $MT(t, r, u, \alpha)$ をマスクドタプルと定義する．ここで t を関係 r 中のタプルとし，u はユーザーを，α はユーザー u に対するアクセス判定関数とする．マスクドタプルは $\alpha(u, PK(t), r.a)$ が false である属性を denied に書き変えたタプルを表す．文脈から明らかなときは単に $MT(t)$ と示す．

定義 4 $MR(r, u, \alpha)$ をマスクドリレーションと定義する．ここで r は関係，u はユーザー，α はユーザー u に対するアクセス判定関数とする．マスクドリレーションはマスクドタプル $MT(t)|t \in r$ をインスタンスとする関係 r を意味する．文脈から明らかなときは単に $MR(r)$ と示すものとする．

定義に示したように，ユーザーが情報にアクセスする権限があるかどうかの判定はアクセス判定関数 α によってなされる．つまり実際にデータベースに格納されている値からアクセス判定がなされるため，データベースの値が変更されるとアクセス判定も自動的に変更される．そのため，RBAC のようにユーザーのロールの変更を行う必要はなくデータベース管理者の負担を大幅に削減できると考えられる．

3.3 アクセスポリシーの記述

RBIM ではアクセスルールの定義に PROLOG や Datalog の要素として知られているホーン節を用いて記述する．前節で述べたアクセス判定関数 $\alpha(s, o, r.a)$ はホーン節で表現されたアクセスルールの集合として定義され，ルールは次のように表される．

$\alpha(s, o, r.a)$:- $condition.$

条件部には述語のリストを記述することで複数の条件を含めることも可能である．以下にアクセスルールの記述例を示す．ここで例に用いる emp テーブルのスキーマは (id, fname, lname, position, salary) とする．

例 1: ユーザー taro はユーザー hanako の emp テーブル中の属性 salary にアクセスできる．

$\alpha(\text{"}taro\text{"}, \text{"}hanako\text{"}, emp.salary)$:- .

例 2: emp テーブル中の属性 fname，lname はだれでもアクセスできる．

$\alpha(X, Y, emp.fname)$:- .
$\alpha(X, Y, emp.lname)$:- .

例 3: 全ての従業員は emp テーブルにある全ての自分の個人情報にアクセスできる．

$\alpha(X, X, emp.*)$:- .

例 4: CEO ならばデータベース中の全ての情報にアクセスできる．

$position(X, P)$:- $emp(X, -, -, P, -)$.
$\alpha(X, Y, *.*)$:- $position(X, \text{"}CEO\text{"})$.

1 行目では emp テーブルの id の値を変数 X に，position の値を変数 P にバインドし，「X の position

```
<people>
  <person>
    <id>1</id>
    <name>
      <fname>Junji</fname>
      <lname>Koizumi</lname>
    </name>
    <salary>10000</salary>
    <position>CEO</position>
  </person>
 <person>
    <id>2</id>
    <name>
      <fname>Jiro</fname>
      <lname>Ozawa</lname>
    </name>
    <salary>7000</salary>
    <position>Manager</position>
  </person>
</people>
```

図 5: サンプルデータ

はP である」ということを定義する述語$position(X, P)$を定義している．また 2 行目の条件部である position 述語は値が CEO である場合のみ true となるため，CEO に対してだけ true を返すアクセス判定関数を定義していることがわかる．

4 XML 問い合わせに対するアクセス制御

本章では 3 章で述べた RBIM の概念を XML 文書に対して適用する手法について述べる．本稿では直接 XML 文書に対してアクセス管理を行う (1) 直接マスキング方式と，XML で文書を関係データベースにマッピングして，関係データベース上でアクセス管理を行う (2) 間接マスキング方式の 2 種類の方式を提案し，以下で詳しく述べる．また，以降の具体例を用いた説明では図 5 の XML データに対する制御を考える．

4.1 直接マスキング方式

直接マスキング方式の概念を表した図を図 6 に示す．システム内で定義されたアクセスルールからユーザーに閲覧が許可された部分だけからなるビューを導出する点では図 4 と全く同じである．

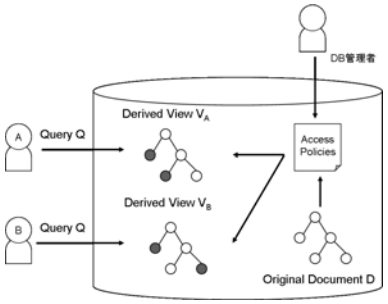

図 6: 直接マスキング方式

ここで関係データベースにおいては 1 タプルが 1 ユーザーの個人情報の単位として扱った．これはユーザーを特定できる属性（主キー）に関数従属性がある属性を個人情報と考えるからである．この概念を XML に応用するにあたり XML Schema[7] のキーの概念を用いる．キーはセレクタとフィールドの 2 つからなり，フィールドは ID の役割を果たしている要素または属性を指し，セレクタはフィールドによって一意に識別できる要素を表している．例えば図 5 のキーを次のように定義すると，id ノードがフィールド，person ノードがセレクタノードとなる．

```
<xsd:key name="person_key">
  <xsd:selector xpath="people/person" />
  <xsd:field xpath="id" />
</xsd:key>
```

本手法ではこのセレクタノードを 1 ユーザーの個人情報の単位と考えることとする．これにより前章で述べた定義を XML に拡張することができる．

4.1.1 RBIM における XML 文書の定義

3.2 節で定義した関係データベースにおける RBIM のモデルを拡張して，以下に XML 文書におけるモデルを示す．

定義 5 XML データベース中の全ての文書 D_i はキー $Key(D_i)$ を持つ．クエリを発行するユーザーのユーザー ID はシステムに登録されており U と表す．

ここでキーのフィールドを $Key(D_i).field$，セレ

クタを $Key(D_i).selector$ と表すものとする．簡単のために，キーのフィールド $Key(D_i)$ は単一のフィールドからなるものとする．

定義 6　$\alpha(s, o, d.xpath)$ はアクセス判定関数というブーリアン関数とする．ここで s はアクセス主体，o はアクセス対象，$d.xpath$ は関係 d 中背クレタノードからの相対パス $xpath$ を表す．アクセス判定関数はあるセレクタノード $n \in d$ において $xpath$ で表される範囲について，ユーザー s がアクセス可能のとき true となる．ここで n は文書 d におけるフィールド $o \in Key(d).field$ によって一意に識別されるものとする．

定義 7　$MN(n, d, u, \alpha)$ をマスクドノードと定義する．ここで n を文書 d 中のセレクタノードとし，u はユーザーを，α はユーザー u に対するアクセス判定関数とする．マスクドノードは $\alpha(u, key(o).field, d.xpath)$ が false である要素や属性を denied に書き変えたセレクタノードを表す．文脈から明らかなときは単に $MN(n)$ と示す．

定義 8　$MD(d, u, \alpha)$ をマスクドドキュメントと定義する．ここで d は文書，u はユーザー，α はユーザー u に対するアクセス判定関数とする．マスクドドキュメントはマスクドノード $MN(n)|n \in d$ をインスタンスとする文書 d を意味する．文脈から明らかなときは単に $MD(d)$ と示すものとする．

4.1.2　アクセスポリシーの記述

アクセスポリシーは 3.3 節で紹介した方式のアナロジを用いて，以下のように定義される．

$\alpha(s, o, d.xpath)$:- $condition$

ここで xpath はセレクタノードからの相対パスを記述するものとする．次にアクセスポリシーの記述例を示す．

例 5: emp ドキュメント中のノード fname，lname はだれでもアクセスできる．

$\alpha(X, Y, emp.name/fname)$:- .
$\alpha(X, Y, emp.name/lname)$:- .

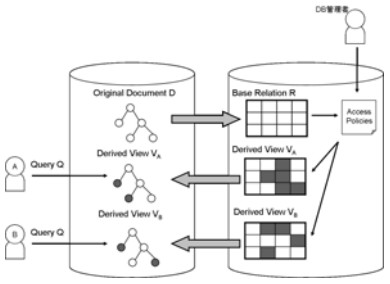

図 7: 間接マスキング方式

表 1: マッピング結果

id	fname	lname	salary	position
1	Junji	Koizumi	10000	CEO
2	Jiro	Ozawa	7000	Manager

例 6: 全ての従業員は emp ドキュメントにある全ての自分の個人情報にアクセスできる．

$\alpha(X, X, emp.*)$:- .

4.2　間接マスキング方式

間接マスキング方式の概念を表した図を図 7 に示す．間接マスキング方式ではユーザーがアクセスしようとしている XML 文書に対して，一度関係データベースにマッピングを行う．そして関係データベース上で RBIM の概念を用いてビューを導出し，そのビューを再度 XML にマッピングを行うことでユーザーに許可された部分だけからなる XML 文書を生成する．

間接方式の利点はインデックスや最適化など長年培われてきた関係データベースの技術を利用できるという点である．図 5 の XML データを関係データベースにマッピングした結果を表 1 に示した．関係データベース上でのアクセスポリシーの管理は前章に述べた定義に従って行うことで，アクセスポリシーからビューを導出することが可能である．また，マッピングをしたときの 1 タプルの情報がユーザーに所有権がある範囲を表すセレクタノード以下の情報に対応しておりアクセスポリシーを定義するときの親和性に優れているといえる．

5 考察

直接マスキング方式における課題としては文書中のどの部分がユーザーに所有権があるのかという議論をより詳細に行う必要がある．今回の提案手法ではセレクタノードの子ノードは全てそのユーザーに所有権があると考えたが，これが妥当であるのか検証する必要がある．また，XML Schema の key-ref のように他の値を参照する場合のノードの所有権などについて考察することを今後の課題とする．

間接マスキング方式の今後の課題はマッピング方式の検討である．例に示したマッピング方式では元の文書の構造に関する情報が失われてしまう点が問題である．これまでにも XML と関係データベースのマッピングの議論は数多くなされているが [5][6]，今回個人情報を取り扱うにあたりデータが誰に所有権があるのかという点を考慮し，文書の構造を保持しつつ所有権の情報も失われないようなマッピング方式を検討する必要がある．

両方式共通の課題として処理時間の削減である．RBIM ではユーザーがクエリを発行した段階でルールに基づいてビューの生成を行うため，処理時間が長くなることが懸念される．特に間接方式では XML から関係データベース，関係データベースから XML へ 2 回のマッピングを行うことによるオーバーヘッドが大きい．今後の課題としてはビューの生成を行う前に部分的にクエリを実行するなど，ビュー生成手順の等価変換による最適化手法について考察を進めていく必要がある．

6 まとめ

本稿では個人情報に対するアクセス制御方式として Rule-Based Information Masking を提案し，XML 文書におけるアクセス制御について述べた．RBIM ではデータベース内のデータに所有者を定義することで，その所有者との間に成り立つアクセスルールに基づいてアクセスが許可された部分だけからなるビューを生成する．また，アクセスポリシーをデータベースシステム内で管理することで，データベースの内容と連動した管理を可能とし，管理者の負担を大幅に削減する．

今後は直接方式，間接方式の両方式の検討および実装を行い，問い合わせに対するレスポンス時間などを両方式で比較していく．

[謝辞] 本研究の一部は，文部科学省の世界的研究教育拠点の形成のための重点的支援 21 世紀 COE プログラム「アクセス網高度化光・電子デバイス技術」の支援によるものである．

参考文献

[1] Oasis. eXtensible Access Control Markup Language (XACML). http://www.oasis-open.org/committees/xacml.

[2] E. Damiani, S. di Vimercati, S. Paraboschi, P. Samarati. A Fine-Grained Access Control System for XML Documents. In *ACM Transaction on Information and System Security*, 5(2):169-202, 2002.

[3] W. Fan, C. Chan, M. Garafalakis. Secure XML Querying with Security Views. In *Proceedings of ACM SIGMOD*, pp.587-598, 2004.

[4] R. S. Sandhu. Role-Based Access Control Models. In *IEEE Computer*, 29(2):pp.38-47, 1996.

[5] M. Yoshikawa, T. Amagasa, T. Shimura, S. Uemura. XRel: a path-based approach to storage and retrieval of XML documents using relational databases. In *ACM Transaction on Internet Technogy*, 1(1):110-141, 2001.

[6] M. J. Carey, J. Kiernan, J. Shanmugasundaram, E. J. Shekita, S. N. Subramanian. XPERANTO: Publishing Object-Relational Data as XML. In *Proceedings of WebDB*, pp105-110, 2000.

[7] World Wide Web Consortium. XML Schema. http://www.w3.org/XML/Schema.