セキュア XML クエリ ――セキュリティビューによるアクセス制御

Secure XML Querying ――Access Control with Security Views

中 山　陽 太 郎

要　約　XML ベースのプロトコルを基盤とする電子商取引や電子申請が普及するに伴い，情報交換におけるセキュリティのリスク管理は，企業にとって重要な問題となってきた．XML セキュリティは，Web サービスのセキュリティフレームワークとして，W 3 C と OASIS によって標準化が進められた．しかしながら，ここで規定されるアクセス制御は，XML クエリにおけるセキュリティモデルとして，必ずしも充分なものではない．本稿では，XML 文書問合せに対する有効なアクセス制御の方法として，セキュリティビューに基づくセキュア XML クエリの手法を紹介する．XML セキュリティビューは，現在エジンバラ大学のデータベースグループで研究されている画期的な手法であり，柔軟なアクセス制御ポリシーの設定と効率的なアクセス制御が特長である．

Abstract　In these days, the spreading of electronic commerce, or electronic application and etc, based on XML-enabled protocol, the enterprise-level risk management in the information exchange has become more important than ever. The standardization of XML security, as the security framework for Web service, has been implemented by W 3 C and OASIS. However, the specification of the access control is not adequate for the XML query process or architecture. In this report, we introduce and explain the outline of the secured XML query technique by access control using security views, which is a novel paradigm studied by the Database Group of University of Edinburgh. The security views approach brings the flexible establishment of the access control policy and the excellent performance for the enforcing of the access control.

1. は じ め に

XML ベースのプロトコルを基盤とする電子商取引や電子申請の普及に伴い，情報交換におけるセキュリティのリスク管理は，企業にとって重要な問題となった．XML セキュリティは，Web サービスのセキュリティフレームワークとして，W 3 C （World Wide Web Consortium）と OASIS （Organization for the Advancement of Structured Information Standard）によって標準化が進められたが，アクセス制御については，XML クエリにおけるセキュリティモデルとして，必ずしも充分なものではない．本稿では，XML 文書問合せに対するセキュリティモデルとして，現在，研究が進められているセキュリティビューに基づくセキュア XML クエリ[3]を紹介する．XML セキュリティビューは，柔軟なアクセス制御ポリシーの設定と，導出されたセキュリティビューによる効率的なアクセス制御が特長である．

2 章で，XML セキュリティの現状と問題点について概観し，RDB や XML　DB のセキュリティの現状について述べる．3 章で，セキュア XML クエリのセキュリティモデルとアクセス制御の仕組みの概略を説明し，4 章で今後の動向について述べる．

2. XML アクセス制御

XML プロトコルを情報共有基盤とする企業システムにおいて，セキュリティ管理は重要な問題である．XML セキュリティは，Web サービスのセキュリティフレームワークとして，W3C と OASIS によって SAML[*1][6]，XACML[*2][7]の標準化が進められてきた．XACML で規定されるアクセス制御は，リソース対象の指定に制約があり，XML 問合せにおけるセキュリティモデルとして充分なものではない．これは，XACML における対象リソースの指定において，XML のノードやアクセスパスによる対象範囲の設定が難しいことに起因する．一方，XML はデータベースと連携して，半構造データモデルとして柔軟なデータ管理を可能にした．データベースにおいて，アクセス制御はセキュリティ上の重要な機能要件であるが，複雑なデータモデルを持つ XML におけるアクセス制御の実現は，RDB に比べ簡単ではない．

ここでは，XACML におけるアクセス制御の概略と問題点について述べ，比較として，RDB 及び XML DB におけるセキュリティを示す．

2.1 Web サービスにおけるアクセス制御の問題点

XML セキュリティの規格として，XML デジタル署名[16]や XML 暗号[10]，鍵管理の XKMS[*3][11]，及び OASIS による Web サービスの相互運用に関連した認証管理に関する SAML，アクセス制御に関する XACML の規定がある．ここでは，Web サービスの相互運用におけるアクセス制御に対するセキュリティモデルを規定した XACML についての概略と，XML 文書問合せの観点から問題となる XACML のリソース指定における規定について述べる．XACML は，現在 XACML 1.0 が標準であり，2004 年 12 月には XACML 2.0 が Committee Draft となっている．

図 1 は，XACML におけるアクセス制御のデータフローを示している．XACML では，アクセス対象リソース（resource）のアクセス権限を定義するために，ポリシー記述言語が規定されており，また，リソースへの実行時の要求を表現するアクセス決定言語がある．リソースの保護ポリシーが検出された場合，リクエストの属性とポリシールール内の属性とを比較し，アクセス許可を決定する．リソース要求をクライアントからサーバに出す場合，アクセス制御を実施するエンティティを PEP （Policy Enforcement Point）と呼ぶ．PEP は，ポリシーを実行するために，要求側の属性を PIP （Policy Information Point）から取得して，PDP （Policy Decision Point）に認可決定を委託する．ポリシーストアに記述された適用可能なポリシーが PDP により評価され，認可決定が返される．PEP では，この情報を使用して適切な応答をクライアントに返すことができる．

図 2 は，XACML のポリシー記述言語の構造を示している．ポリシーは，対象（Target），ルール（Rule），債務（Obligation）の各要素から構成され，ルールには条件（Condition）を付加できる．複数のポリシーまたはポリシー集合を結合して一つのポリシー集合 （PolicySet）を作る．主体（Subjects）は，要求コンテキストで示される主体の属性のルールを記述する．リソース（Resource）は，要求コンテキストが示すリソース属性に対して，ルールを適用する対象を限定し，アクセスの対象を指定する．動作（Action）は，要求コンテキストが示す動作属性に対して，ルールを適用するリソースへのアクセスに対する動作を指定する．SAML で定める Read, Write, Delete などの動作に加えて XACML では任意の動作も定義することができる．

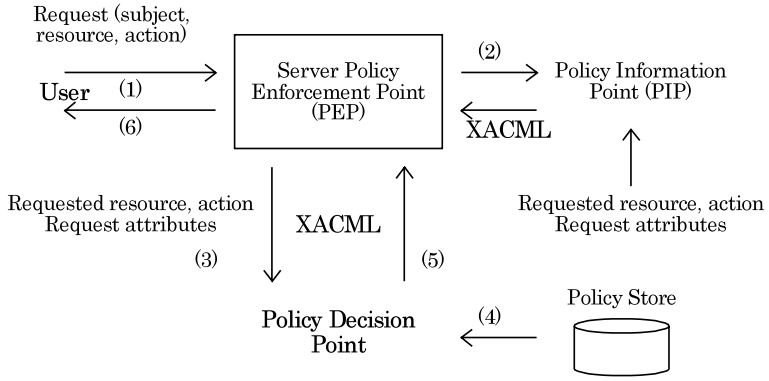

図 1　XACML アクセス制御のデータフローモデル

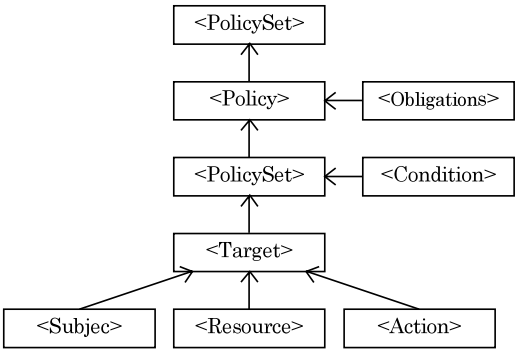

図 2　ポリシー記述言語の構造

XACML におけるルール適用のリソースの限定は，資源識別子による XML データの外部実体レベルに留まっているのが現状であり，リソースの内部実体レベルにおけるアクセス制御の対象範囲を内部構造に即して明示的に指定するには限界がある．アクセス制御の対象範囲を内部実体レベルまで拡張し，スキーマで規定される XML 文書の内部構造に即した整合性のある設定を可能にするのが，本稿で紹介するセキュリティビューの利点の一つである．

2.2　RDB のアクセス制御

標準 SQL[*4] では，セキュリティに関する規約として権限管理とロールが規定されている．権限は，データベースオブジェクトに対する特定のアクセス権限であり，ロールは，いくつかの権限を纏めたものである．設定可能な権限としては，SQL のデータ定義，及びデータ操作（SELECT, INSERT, UPDATE, DELETE）に関するものであり，カラムや行単位におけるアクセス制御については規定されていない．

DB サーバのセキュリティモデルとしては，標準 SQL で規定される権限管理だけでは，不十分であり，代表的な商用 RDB である Oracle 10 *g* では，表 1 に示す統合的なセキュリティ機能を提供している．

表 1　Oracle 10 g のセキュリティ機能

セキュリティ項目	Oracle セキュリティ機能
認証管理	Oracle Identity Management，グローバル認証，外部認証，プロキシ認証，DB 認証，OS 認証
通信データの暗号化	Advanced Security，パスワード暗号化
アクセス制御	仮想プライベート DB
格納データの暗号化	暗号化ツールキット
監査	標準監査，DBA 監査，ファイングレイン監査，イベントトリガー，ログマイナー

認証管理，通信データの暗号化はセッションや通信におけるセキュリティ管理であり，アクセス制御，データ暗号化，監査は，サーバ側のセキュリティ管理である．Oracle におけるアクセス制御は，仮想プライベート DB （Virtual Private Database ： VPD）と呼ばれ，オブジェクト権限による表単位でのアクセス制御に加え，行単位でのアクセス制御を可能にする．VPD は，SQL 文を動的に変更するファイングレイン　アクセスコントロールとユーザセッション情報を管理するアプリケーションコンテキストの二つの要素から構成され，アクセス制御ポリシーはメタデータとして管理される．

2.3　XML DB のアクセス制御

XML データベースは，インターネットの普及と伴に Web 上の柔軟なデータ構造を扱う半構造データベースとして研究が進められてきた[1]．現在 XML DB としては，ネイティブ XML DB と RDB をベースとする XML　DB の二つがあり，XML 対応 DB 製品も多数存在する．しかしながら，XML　DB としてのセキュリティ機能に関しては，RDB と比較した場合，まだ充分とは言えないのが現状である．以下では，XML DB 製品の例として Tamino と Oracle XML DB を取り上げる．

Tamino[8]は，階層型 DBMS をベースとしたネイティブ XML DB の商用プロダクトであり，XPath[12]，XQuery[17]，XML　Schema[13][14][15]等の規格をサポートしている．アクセス制御については，インスタンス単位でアクセス権限を設定することが可能であるが，内部データと同期したアクセス制御の管理には限界があり，セキュリティの整合性が問題である．

Oracle XML DB[5]は，XML Schema, XPath 等をサポートするネイティブ XML DB である．セキュリティに関して，アクセス制御リストに関するサポートがあるが，これは WebDAV（Distributed Authoring and Versioning）の仕様[*5] に準拠した認証と操作に関するロールベースのアクセス制御を行うもので，XML データ内部に対するアクセス権限まで制御するものではない．

XML　DB では，RDB プロダクトで実現されているセキュリティフレームワークと同等の機能が充分に提供されていないのが現状である．これは，XML 文書が関係データベースのデータ構造に比べ，複雑で扱いにくいことに起因している．XML の自己記述的で複雑なデータ構造において，アクセス権限を管理することは，関係モデルに比べて複雑なものになる．XML DB においては，XML データ操作としての XQuery, XPath 等の規格が普及し始めた段階であり，セキュリティ機能については，本稿で紹介するセキュア XML クエリを含め，まだ検討の段階にあるといえる．

3. セキュリティビューによるセキュア XML クエリ

セキュリティビューによるセキュア XML クエリは，XML 文書の問合せに対するアクセス制御をベースとしたセキュリティ技術である．XML スキーマレベルでセキュリティポリシーを定義することが可能であり，アクセス制御対象となるリソース範囲を明示的に規定することができる．また，セキュリティビューの適用により，アクセス制御の実施に伴う効率の低下を抑制することが可能である．この章では，セキュリティビューで実現される XML 問合せにおけるセキュリティモデルとその仕組みの概略について説明する．

3.1 セキュリティビューの概要

XML データの実体レベルに対するアクセス権限の指定は，セキュリティ管理の面で大きな負担となると同時に，アクセス制御の実施においても，データアクセス処理の大きなオーバーヘッドとなる．セキュリティビューによるアクセス制御では，スキーマレベルのアクセスポリシー定義と XML 文書スキーマから導出するセキュリティビューによってアクセス制御を実現する．XML 問合せにおけるアクセスポリシーの定義が，XML スキーマレベルで適用できることは，セキュリティ管理とアクセス制御の実施において極めて有効である．スキーマレベルによるセキュリティ管理によって，機能的に以下の利点が得られる．

- アクセス可能データを定義したセキュアスキーマ（セキュリティビュー）の提供
- セキュリティ管理と XML 文書インスタンスの分離：インスタンスレベルでのセキュリティ管理不要

セキュリティビューでは，スキーマレベルのセキュリティポリシーとクエリ書換えによるアクセス制御によって，セキュリティ管理と実データの分離が保障される．XML スキーマに対するセキュリティポリシーの定義によって，セキュリティの保守性が向上し，データやスキーマの変更に対しても，整合性を保持しつつ柔軟に対応できる．また，セキュリティの実施においても，セキュリティビューによりアクセス不可データは隠蔽され，同時に書換えによるクエリの実行により，データアクセス時のオーバーヘッドは最小限に抑制される．

図 3 は，セキュリティビューによるセキュア XML クエリの概要を示している．ユーザまたはグループごとにアクセス権限を規定したセキュリティポリシーが定義される．ユーザは，XML 文書または XML データベースへの同時アクセスが可能であり，データアクセス時にセキュリティポリシーに記載されたアクセス権限に応じたアクセス制御が施行される．アクセス制御ポリシーから生成されるセキュリティビューを通してデータにアクセスすることにより，アクセス権限の認められた参照可能な XML ノード及びサブツリー以外は隠蔽される．

3.2 セキュリティビューによるアクセス制御モデル

XML 文書のアクセス制御は，アクセス制御ポリシー仕様言語，アクセス制御ポリシーからのセキュリティビューの生成，セキュリティビューに適合するクエリの書換えによるアクセス制御の実施によって実現される．セキュア XML クエリでは，構成要素として，アクセス権限を定めたセキュリティポリシー，セキュリティビュー導出モジュール，クエリ変換モジュールを持ち，それぞれ以下の役割を担う（図 4）．

- アクセス制御ポリシー定義（Access Control Policy Specification）：DTD 文書とそれに対する XPath 修飾で注釈付け（annotation）することにより，XML 文書に対してスキ

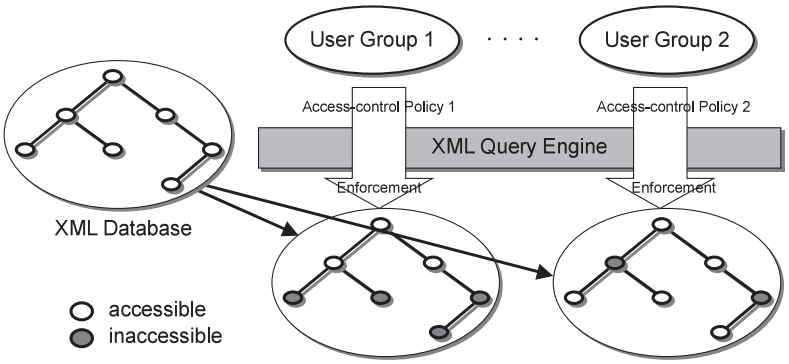

図 3　アクセス制御ポリシーの施行

ーマレベルで，そのアクセス可能範囲を規定する．

- セキュリティビュー導出モジュール（Security View Derivation Module）：各ユーザまたはグループごとのアクセスポリシーから，対象 DTD に対するセキュリティビューを導出する．セキュリティビューは，参照可能な XML 文書の範囲が規定された DTD 文書である．ユーザは，セキュリティビューに対して，クエリを発行する．
- クエリ変換モジュール（Query Translation Module）：セキュリティビューに基づくユーザのクエリを，実際の DTD 文書に対する等価なクエリへの書換え，及び最適化を行う．

ユーザは，セキュリティビューとして提供される DTD に対するクエリを発行することで，クエリ変換モジュールは，実際の DTD 文書に対して，アクセス制御条件を満たす等価なクエリへの書換えを実行する．クエリエンジンは，書換えられたクエリを使用して，実際の XML データに対する問合せ評価を行う．特にクエリ書換えの利点としては，アクセス可能なデータセットを生成することなく，クエリーレベルでアクセス可能なデータの制御が可能なことである．

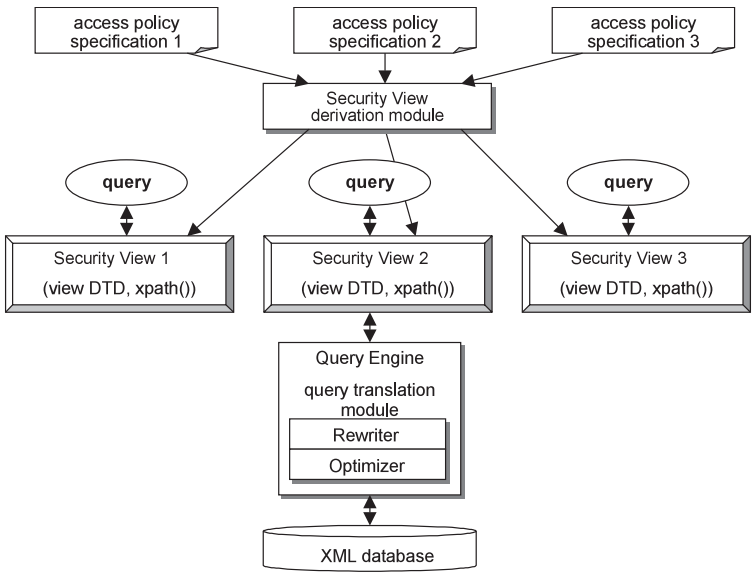

図 4　セキュリティビューによるアクセス制御モデル

このように，アクセス制御ポリシーによるセキュリティ管理では，スキーマとして DTD を，対象リソースの範囲の限定に XPath による注釈を使用してセキュリティビューを導出し，セキュア XML クエリを実現している．

以降の章では，アクセスポリシーの定義，セキュリティビューの導出，及びクエリ書換えとアクセス制御の実施について，それらの定義の概略とそれぞれの特長について述べる．

3.3 アクセスパスの記述

XML 文書型定義 DTD は，XML 1.0[9]で規定された XML スキーマ言語の一つである．文書型定義 DTD D の要素タイプ DTD $D : A \rightarrow \alpha$ は次のように定義できる．

$$\alpha ::= PCDATA \mid \varepsilon \mid A_1, \ldots, A_n \mid A_1 + \ldots + A_n \mid A^*$$

ここで，*PCDATA* は文字データ，ε は空要素，“ , ”，“ ＋ “ 及び “ ＊ ” はそれぞれ連結，選言，繰返しを示している．XML 文書は，DTD に対して妥当性検証が行われなければならない．

XPath[12]問合せは，XML データの階層構造から，パス指定によって要素や属性などのデータを辿ることにより，抽出するための記述方法を規定した仕様である．XPath では，XML 文書上のノードの位置を指定するために，さまざまな式（Expression）を記述できる．XPath 問合せは，次のように定義される．

$$p ::= \varepsilon \mid l \mid * \mid p/p \mid //p \mid p \cup p \mid p\,[q]$$

ここで，ε，l 及び＊は，それぞれ空パス，ラベル，ワイルドカードであり，“ / ”，“ // ”，及び “ ∪ ” は，それぞれ子ノード，子孫ノード，和を示している．p［q］は限定子（Qualifier）と呼ばれ，次のように定義される．

$$q ::= p \mid p = c \mid q \wedge q \mid p \vee p \mid \neg q$$

ここで c は定数，p は上で定義されたものである．また，“ ∧ ”，“ ∨ ”，及び “ ¬ ” は，それぞれ連言，選言，否定である．

XPath 問合せ q は，XML 文書におけるコンテキストノード（現在の位置）v において評価され，v から p へのアクセス可能なノードの集合（または文字データ）v〚p〛を結果として返す．限定子 p［q］は，*XML* 文書におけるコンテキストノード v において，v〚p〛が空で無い場合に限り，［p］が成り立つものとする．

3.4 アクセス制御ポリシー定義

アクセス制御ポリシーは，アクセス制御の仕様定義であり，対象となる DTD 文書とそれに対するアクセスパスを定義する XPath によって規定される．アクセス制御ポリシー S は，S ＝（D，*access*（　））として定義され，写像 *access*（　）は，DTD 文書のノードに対するアクセス属性（Accessibility）をアクセスの可否として指定し，*access*（　）：＝Y | N | ［q］として定義される．*access*（A，B）は，要素 A から子要素 B に対するアクセス属性への関数であり，Y，N，［q］はそれぞれ，アクセス可，アクセス不可，条件付きアクセス可（条件［q］を満たす場合アクセス可） であることを示している．アクセス制御ポリシーは，ポリシー定義言語として，以下の特性を備えている．

1） オーバーライド（Override）：*access*（A，B）＝Y の場合，A の子 B は，A のアクセス属性をオーバーライドする．

2) 継承（Inheritance）：*access*（*A*, *B*）が明示的に定義されていない場合，A の子 B は，A のアクセス属性を継承する．
3) コンテンツベース（Content-based）：XPath 限定子による条件アクセス属性．
4) DTD 文書の XML ツリー：各データノードのアクセス属性は，アクセス定義によって一意に定義される．ノードは，文書ルートから関連するパスに対して，及びノードに対する限定子がそのノード以下の部分木を束縛することによって決定される．
5) セキュリティの対象粒度：特定のノード，及びサブツリーが指定可能．
6) データ定義レベル：セキュリティの設定として，XML 文書のインスタンスレベルに対する指定は不要である．これによって，効率に優れ，容易なアクセス定義や保守が実現できる．

例として，hospital データベース（図 5）の DTD に対するアクセス制御ポリシーを検討する．まず，ユーザ *nurse* の hospital データベースに対するアクセス制御ポリシーを定義する．ユーザ *nurse* は，trial, trName, regular, bill の各要素に対するアクセス権限を持たず，patient, SSN, name, record, date, diagnostic, tname の各要素に対しては，［q 1］，［q 2］で示される条件付でアクセス可能とする．図 5 の DTD グラフに×印を付けて示したノードはアクセス不可を示す．

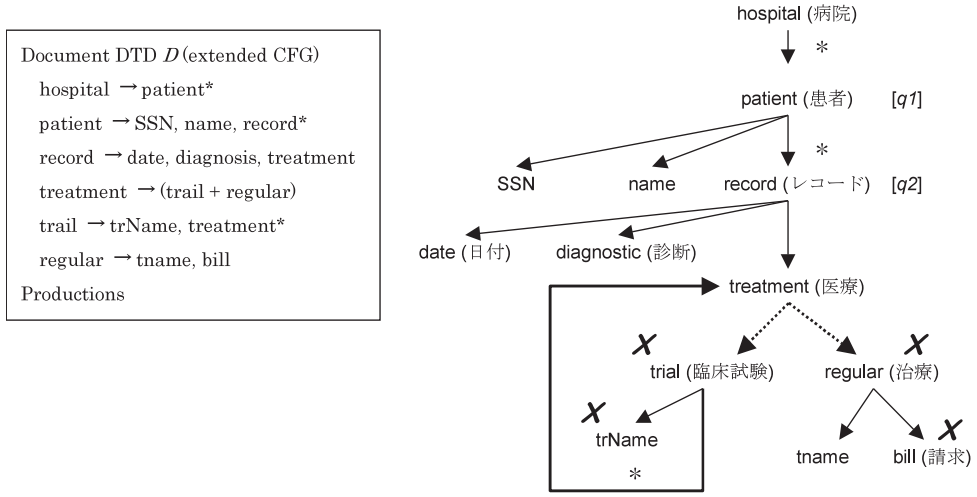

図 5　hospital データベースの DTD グラフ

ユーザ *nurse* の hospital データベースに対応するアクセス制御ポリシー定義を表 2 に示す．ポリシーは，DTD グラフの辺（edge）に対して，アクセスの可否を定義したものである．patient, SSN, name, record, date, diagnostic, tname は，［q 1］，［q 2］で示される条件付でアクセス可能であり，ssn, name は，親ノード record のアクセス条件を継承する．trial, regular は明示的にアクセス不可が定義され，trName, bill は，アクセス不可を親ノードより継承する．

3.5　セキュリティビューの生成

セキュリティビューは，アクセス制御ポリシーから生成され，DTD 文書のインスタンス D から view DTD のインスタンス D_v への写像として定義される．アクセス制御ポリシーを S ＝

表 2　アクセス制御ポリシー定義

hospital → patient*
　　access (hospital, patient) = [//diagnose = "*DIS*"]　・・・ [q1]
patient → record*
　　access (patient, record) = [diagnose = "*DIS*"]　　・・・ [q2]
treatment → trial + regular
　　access (treatment, trial) = *N*
　　access (treatment , regular) = *N*
regular → tname
　　access (regular, tname) = *Y*

（*D*, *access*（　））とすると，アクセス定義 *S* から view DTD D_v へのセキュリティビュー *V* : *S*→D_v は，*V*＝（D_v, *xpath*（　））として表すことができる．*xpath*（　）は XML 文書からアクセス可能なデータを抽出するための XPath 問合せの注釈（annotation）を示している．ここで *A*, *B* を DTD 文書の要素とすると，DTD 文書 *D* に対する XPath 問合せ *xpath*（*A*, *B*）は，DTD 文書からデータを抽出して，要素 *A* に対する子要素 *B* を生成することを表す．要素がルートノードの場合は，*xpath*（r_v）で表される．

アクセス制御ポリシー *S*＝（*D*, *access*（　））に対するセキュリティビューは，*V*＝（D_v, *xpath*（　））で与えられる．ここで，例として表 3 に，前節で示した hospital データベース（図 5）のセキュリティビューを示す．

表 3　セキュリティビュー

production:　hospital → patient*
　　xpath (hospital, patient) = hospital/patient [q1]
　　[q1]: [//diagnose = "*DIS*"]
production:　patient → SSN, name, record*
　　xpath (patient, SSN) = SSN, /* name */
　　xpath (patient, record) = record [q2]
　　[q2]: [diagnose = "*DIS*"]
production:　record → date, diagnosis, treatment
　　xpath (record, date) = date, /* diagnosis */, /* treatment */
production:　treatment → tname*
　　xpath (treatment, tname) = //tname

図 6 では，元となる DTD 文書 *D* に対してアクセス制御ポリシーが適用された結果，セキュリティビューが生成され，View　DTD 文書 D_v では元の DTD 文書 *D* では表示されていた trial, trName, regular, bill が隠蔽され，アクセス可能なノードのみ現れていることを示している．

セキュリティビュー導出モジュール（図 4）における入力と出力は以下のとおりである．

　入力：アクセス制御ポリシー　*S*＝（*D*, *access*（　））

　出力：セキュリティビュー定義　*V*＝（D_v, *xpath*（　））

セキュリティビューの派生は，DTD に対してトップダウンに行われ，与えられたアクセス制御ポリシーに対して，セキュリティビューを導出する．T_v の導出が終了すると，セキュリティビュー *S* に対するアクセス可能な *T* の要素群が構成される．*S* に関するそのようなアクセス制御ポリシーが存在する場合に限り，*S* に関して *V* は完全であるという．

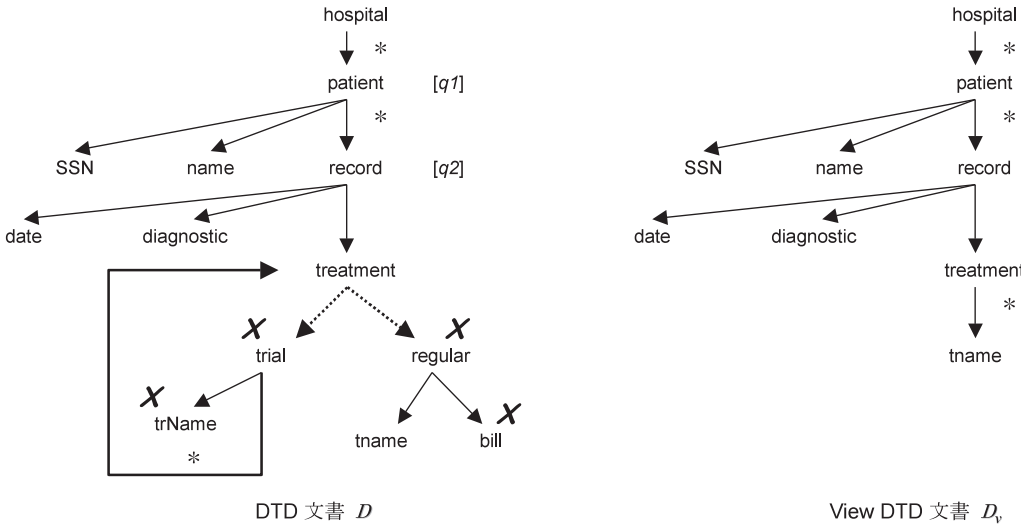

図 6　セキュリティビュー　view DTD

セキュリティビュー派生アルゴリズムでは，V＝（D_v, *xpath*（　））を構成するときに，ショートカット（*short-cutting*）とノードの再命名（*renaming*）によって DTD D のアクセス不可ノードを隠蔽する処理を行う．アルゴリズムの詳細は省略するが，セキュリティビュー生成の計算量は O（｜D｜^{2}）（但し｜D｜は，DTD 文書のサイズ）で与えられる．

セキュリティビュー D_v によって，元となる DTD 文書における，適切（relevant）かつアクセス可能な部分の構造と意味を保持することがセキュリティビュー派生アルゴリズムの要件である．

3.6　クエリ書換えによるアクセス制御の実施

ここでは，セキュリティビューによる XML クエリについて述べる．S＝（D, *access*（　））かつ V＝（D_v, *xpath*（　））であるようなセキュリティビュー V ： $S \rightarrow D_v$ を考えてみる．セキュリティビューによるクエリ実施の方法としては，DTD 文書 D のインスタンス T が与えられた場合，D_v のインスタンス Tv を計算し，Tv に対するクエリを評価する方法がある．しかしながら，この方法では，文書の実体化のオーバーヘッドとビューのメンテナンスに問題がある．セキュリティビューに対するアクセス制御実施のためには，クエリ書換えが有効である．セキュリティビューに対するクエリ Q が与えられたとする．セキュリティビュー DTD D_v に対するクエリ Q は，DTD D に対するクエリ Q_t として書き換えられる．つまり，D のインスタンス T に対して，以下が成り立つ．

$$Q_t(T) = Q_v(V(T))$$

XML アクセス制御仮想システム（図 4）で与えられる問合せ評価モジュール（Query evaluation module）の書換え処理（rewriter）における入力と出力は以下のとおりである．

入力：セキュリティポリシー　S＝（D, *access*（　））に対するセキュリティビュー　V＝（D_v, *xpath*（　））

及び，セキュリティビュー　V＝（D_v, *xpath*（　））に対する XPath クエリ Q_v

出力：DTD D に対する XML 文書 T に対する XPath クエリ Q_t

例として，再び hospital データベース（図 5）におけるクエリを考える．

セキュリティビュー V のインスタンス V（T）に対するクエリ

Q_v＝//*patient*［*name*＝“*Joe*”］//*tname*

⇔ xpath（*hospital*, *patient*）［name＝“*Joe*”］/
xpath（*patient*, *record*）/
xpath（*record*, *treatment*）/
xpath（*treatment*, *tname*）

⇔ ドキュメント D のインスタンス T に対するクエリ

Q_t＝/*hospital*/*patient*［*name*＝“*Joe*”and//*diagnosis*＝“*DIS*”］
/*record*［*diagnosis*＝“*DIS*”］
/*treatment*//*tname*

これは，セキュリティビュー V に対するクエリ Q_v がドキュメント D のインスタンス T に対するクエリ Q_t と等しい問合せ結果を返すことを示している．問合せ評価モジュールの書換え処理において，書換え（*rewrite*）アルゴリズムにより，Q_t（T）＝Q_v（V（T））を満たすクエリ Q_t が生成される．

4. セキュリティビューの今後の動向

セキュリティビューは，XML スキーマレベルでセキュリティ定義と実施を可能にしたセキュリティモデルであり，以下のような特質を持つ．

- ファイングレイン・アクセス制御定義言語
- セキュリティビューによる実施方法の効率化

セキュリティビューのファイングレイン・アクセス制御の方法は，2.2 節で示した RDB のアクセス制御方法とクエリ書換えを行うという点で近いものである．また，今回取り上げたセキュリティビューによる方法の他にも XML 文書に対するファイングレイン・アクセス制御に関する研究[2]があるが，これは XML 文書からビューを生成するアプローチであり，スキーマを利用する方法によるものではない．XML 文書を直接管理する方法は，スキーマレベルでの管理に比べ，セキュリティ管理が複雑になり，またアクセス制御実施における実行効率の点で問題がある．

XML セキュリティビューの優れた点としては，リレーショナルの単純なデータモデルではなく，XML の複雑なデータモデルにおいて，仮想ビューが実現できることを示した点にある．また，これによって，クエリ評価モジュールレベルに組み込むことが可能となり，基本的にデータベースのエンジン部分に対する拡張は不要である．セキュリティビューは，アクセス制御での用途に留まらず，XML ビューとして広く一般的な機能として応用されることが期待される．セキュリティビューで提示された方法が，今後多様な XML の規格に対して，どのように適用，または応用されていくのか注視すべきである．

5. お わ り に

本稿では，セキュリティビューによるセキュア XML クエリを紹介した．セキュリティビューは，仮想ビューによって XML アクセス制御を実現する画期的な手法であり，現在，エジンバラ大学のデータベースグループによって研究が進められている．セキュリティビューによる XML 問合せ処理は，XACLM における XML 文書ベースのアクセス制御に比べ，ノード，ア

クセスパスの柔軟な設定，及びアクセス制御の実施の点で優れている．今回紹介したセキュア XML クエリは，XPath の一部に限定されたものであるが，今後，セキュリティポリシーに基づく書換えと最適化の手法を XSLT や XQuery にまで拡張すること，及びセキュリティビューとして DTD のみではなく，XML Schema をサポートすることなどが検討されている．

なお，本稿では，クエリ評価の最適化，アルゴリズムとその効率性，及び理論的側面についての説明は割愛させていただいた．セキュリティビューは，実システムへの応用のみでなく，理論的にも興味深く，計算量や完全性の問題も取り上げられている．関心ある読者は，ぜひ論文[3]を参照されることをお勧めする．

---

*1 Security Assertion Markup Language．セキュリティ情報（認証，属性，認可）の記述とやりとりに関する標準を規定する仕様．
*2 Extensible Access Control Markup Language．セキュリティ情報のうち，認可制御ポリシーの記述と認可要求処理サービスに対するインターフェイスに関する標準を規定する仕様．
*3 XML Key Management Services．公開鍵基盤（PKI）を使用した公開鍵情報の取得や登録などを Web サービスとして使用可能にするための仕様．
*4 標準 SQL としては，SQL : 2003 が最新である．主に Java 言語，及び XML 文書対応の拡張が行われている．
*5 WebDAV （Web-based Distributed Authoring and Versioning）は，RFC 2518 で定義されているプロトコルの名称であり，Web コンテンツの編集やリビジョン管理を目的としたものである．

参考文献

［1］ S. Abiteboul, P. Buneman, and D. Suciu,“ Data on the Web. From Relations to Semistructured Data and XML ”, Morgan Kaufman, 2000.
［2］ E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati, “ A Fine-Grained Access Control System for XML Documents ”, ACM TISSEC, 2002.
［3］ W. Fan, C. Chan, and M. Garofalakis, “ Secure XML Querying with Security Views ”, ACM SIGMOD, 2004.
［4］ M. Graff and K. van Wyk,“ Secure Coding Principles and Practices ”, O ’Reilly, 2003.
［5］ M. Drake, “ Oracle XML DB White Paper ”, Oracle Corporation, 2004.
［6］ OASIS, Conformance Requirement for the OASIS Security Assertion Markup Language（SAML）V 2.0, Committee Draft, December, 2004.
［7］ OASIS, eXtensible Access Control Markup Language（XACML）, Committee Draft, December, 2004.
［8］ Tamino XML Server, Tamino Security バージョン 4.1.5，操作手順書，BeaconIT, 2003.
［9］ World Wide Web Consortium（W 3 C），XHTML（tm）1.0 The Extensible HyperText Markup Language（Second Edition），August, 2002.
［10］ World Wide Web Consortium（W 3 C），XML Encryption Syntax and Processing, December, 2002.
［11］ World Wide Web Consortium（W 3 C），XML Key Management Specification （XKMS 2.0）Version 2.0, Candidate Recommendation, April, 2004.
［12］ World Wide Web Consortium（W 3 C），XML Path Language（XPath），August, 2002.
［13］ World Wide Web Consortium（W 3 C），XML Schema Part 0 : Primer, May, 2001.
［14］ World Wide Web Consortium（W 3 C），XML Schema Part 1 : Structures, May, 2001.
［15］ World Wide Web Consortium（W 3 C），XML Schema Part 2 : Datatypes, May, 2001.
［16］ World Wide Web Consortium（W 3 C），XML-Signature Syntax and Processing, February, 2002.
［17］ World Wide Web Consortium（W 3 C），XQuery 1.0 : An XML Query Language, Working Draft, October, 2004.

執筆者紹介　中 山　陽 太 郎（Yotaro Nakayama）
1988 年東京学芸大学大学院教育学修士課程修了．同年日本ユニシス(株)入社．HMP　IX シリーズ　データベース管理システムの開発保守に従事．現在ユニアデックス(株)ソフトウェアプロダクト統括部サーバソフトウェア二部所属．