アクセス制御機構を有するセキュア WebDAV の開発

電子政府等の大規模組織に適したドキュメント共有・協調編集システム

松原克弥[1)]　端山貴也[2)]　北川和裕[3)]　桝田義一[4)]

1)　株式会社 イーゲル matsu@igel.co.jp

2)　株式会社 イーゲル taki@igel.co.jp

3)　慶應義塾大学 政策メディア研究科 kaz@w3.org

4)　株式会社 SRA 先端技術研究所 masuda@sra.co.jp

概要:

インターネットで広く使われている HTTP 通信プロトコルを拡張した WebDAV は、地理的に離れた複数のユーザによるドキュメント等の協調作成および共有のためのプラットフォーム技術として注目されている。WebDAV 基本プロトコルは、プロトコル仕様にアクセス制御のための機構を定義していない。既存の WebDAV システムの多くは、HTTP ユーザ認証等の外部技術を組み合わせることでアクセス制御の実現を行っているが、設定が細かく管理が煩雑となり、大規模なシステムには適用が難しい。本開発では、企業や政府で使われるような大規模システムのアクセス制御メカニズムとして注目されているロールベース･アクセス制御（RBAC）を用いて、アクセス制御機構を持つセキュア WebDAV システムの設計および実現を行った。

1　背景

インターネット上で文書共有に HTTP が広く用いられている。WebDAV は、HTTP を拡張したもので、地理的に分散したユーザによるドキュメントの共有および協調オーサリングのための機能を提供する。WebDAV は、分散オーサリングシステムの主要なサービスとしての地位を獲得しつつあり、情報基盤のひとつとしての活用が期待されている。

しかしながら、現在の WebDAV は、基本的にセキュリティの確保を意識しておらず、ファイルの読み込み、更新、消去などの操作に関する許可・不許可の制御も行われていない。従って現状では電子政府等の機密性を要する文書を取扱う環境での利用には向かない。

WebDAV のセキュリティ向上策のひとつとして、現在、IETF においてアクセス制御の機能を提供するプロトコル（WebDAV ACL）の標準化が進められている。WebDAV ACL は、従来一般的なタイプのアクセス制御機能を提供するが、抽象度が低く柔軟性を欠くため、ユーザの利用環境に対応させ難い。また、文書のアクセス制御機能のみが提供され、ユーザ管理や権限の委譲については別途機能を必要とする。

他方、近年、アクセス制御技術の分野では、ユーザの役割に応じたアクセス制御（RBAC: Role Based Access Control）が注目されている。従来一般的なアクセス制御方式に比べ、RBAC はより高次元で抽象度の高いアクセス制御が可能であり、下位のシステムとは独立したシステムとして構成できる。RBAC は既にいくつかの商用システムで採用され、今後さらに普及と改良が進むと考えられる。電子政府のように、役割に応じた権限が与えられている組織や機関で、担当者が頻繁に入れ替わるような利用場面においては RBAC によるアクセス制御が有効である。

2 開発の目的

本開発では、WebDAV のセキュリティおよびアクセス制御機能の課題を解決するために、現在利用されている暗号機能 SSL をセキュアな通信路として利用した、WebDAV に対して RBAC を用いたアクセス制御機能の設計と実装について述べる。この設計・実装においては、ユーザのアクセスモデルには、RBAC モデルを用いるが、WebDAV に RBAC モデルに基づいた ACL 機能の通信プロトコルとして、WebDAV を拡張した WebDAV ACL を利用した。

開発においては、RBAC 認証システムを独立したサーバとして実装した。さらに、RBAC インタフェースおよびデータは、WebDAV ACL との整合性を高めるため XMLSchema で規定している。従って、任意の XML を支援している任意のアプリケーションおよびサーバから利用することができる。

また、WebDAV に RBAC を導入するために、既存の WebDAV プロトコルの拡張を試みた。WebDAV はセッションレスのシステムであるが、RBAC は逆にセッションを意識したシステムを想定している。従って、二つのシステムのギャップを埋め、整合性をたかめるため RBAC を WebDAV に応用する際に既存の WebDAV プロトコルの拡張を行った。

上記の WebDAV RBAC サーバ、クライアントおよび RBAC サーバそれぞれを実現した。実現した WebDAV RBAC サーバは、既存の RBAC 機能を保持しない WebDAV クライアントからでも RBAC 機能を用いた利用が可能である。従って、汎用 WebDAV クライアントからも、RBAC に基づいたアクセス制御を利用可能な実装を実現している。

3 設計

3.1 システムアーキテクチャ

本開発では、アクセス制御情報を管理する RBAC システムと RBAC システムの提供する情報を用いて WebDAV リソースのアクセス制御を行う WebDAV RBAC モジュールを組み込んだ WebDAV システムの 2 つのシステムに分離する設計を行った。RBAC システムと WebDAV システムを分離することによる利点を以下に示す。

- インターネットからの不特定多数のアクセスにさらされる WebDAV システムから RBAC 管理情報を分離することで、クラッカによる攻撃から RBAC 管理情報をより保護し易い構成が可能となる。

- RBAC システムを独立にすることで、さまざまな WebDAV 実装システムに適用することが容易になる。

- 複数の WebDAV システムをひとつの RBAC システムで一元的にアクセス制御管理を行うことができる。

- RBAC システムを WebDAV 以外のシステムへ流用することができる。

本設計で構築するシステム構成を図 1 に示す。各モジュールは、以下のような役割を果たす。

WebDAV RBAC クライアント：

RBAC による認証、資源アクセスの許諾を行う WebDAV クライアント。

WebDAV RBAC サーバ：

RBAC を用いた認証セキュリティ機能を有する WebDAV サーバ。WebDAV RBAC クライアントに資源アクセス機能を提供する。RBAC サーバを用いてユーザの認証、資源アクセスの許諾を行う。

RBAC サーバ：

ユーザの情報、ロール、パスワードの管理を行う。また、資源のアクセス権の管理も行う。RBAC データベースのフロントエンドとして実現され、様々なサーバと通信可能。

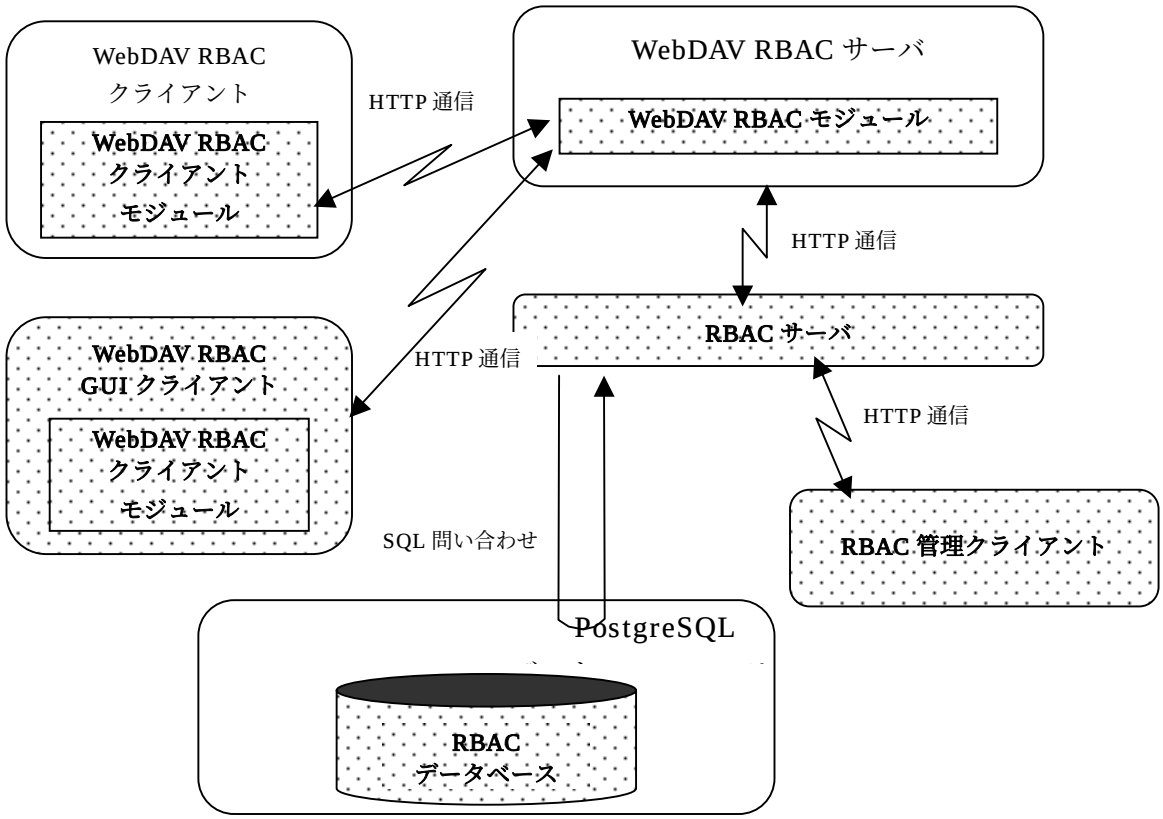

図1　システムアーキテクチャ

RBAC データベース：

RBAC に関するデータの管理、資源のアクセス権に関するデータの管理を行う。RBAC サーバのためのバックエンドデータベース。RBAC サーバが管理するユーザおよびロールに関するデータを管理する。

クライアントが WebDAV サーバ上の資源にアクセスする場合、WebDAV サーバはクライアントから送られた、ユーザ名、ロール名、パスワードを基に認証を行い、資源アクセスの許諾を行い、要求された資源をクライアントに提供する。

実際の処理の流れは、次のようになる。

1. クライアントから WebDAV RBAC サーバに資源の要求を出す。

2. クライアントが RBAC 機能を持つ場合、RBAC サーバにユーザ名、ロール名、パスワードを送る。

3. WebDAV RBAC サーバは、クライアントから受け取った情報を RBAC サーバに送り、ユーザ認証を行う。ユーザが認証された場合は、RBAC サーバは、WebDAV RBAC サーバにセッション ID を送る。

4. WebDAV RBAC サーバは、クライアントにセッション ID を送る。以後、クライアントは、セッション ID を基に WebDAV の資源にアクセスする。

5. WebDAV RBAC サーバは、RBAC サーバと共同して資源アクセスの許諾を行う。

6. 資源アクセスが許可された場合、クライアントから要求された操作を資源に対して行う。

3.2 プロトコル

複数のサーバから構成される本システムでは、WebDAV クライアントと WebDAV RBAC サーバ間および WebDAV RBAC サーバと RBAC サーバ間の通信プロトコルを設計する必要がある。

3.2.1 RBAC プロトコル

RBAC プロトコルは、RBAC 仕様に規定されプリミティブ呼び出しとそのレスポンスを記述可能としなければならない。本プロトコル設計では、XML を用い、リクエストとその属性を記述可能とした。また、ヘッダには、ユーザの認証方式および認証情報を示すタグを定義した。プロトコルの例を以下に示す。

```
<?xml version="1.0" encoding="utf-8" ?>
<Rbac>
  <RbacHdr>
    <Version>1.0</Version>
    <Auth>
      <HTTPBasicAuth>
        <Realm>davserv.org</Realm>
        <Algorithm>b64</Algorithm>
        <PassPhrase>hhifsdifhds==</PassPhrase>
      </HTTPBasicAuth>
    </Auth>
  </RbacHdr>
  <RbacBody>
    <Method>CreateSession</Method>
  </RbacBody>
</Rbac>
```

上記プロトコルで記述可能な CoreRBAC 基本 API は、以下の 21 個である。

Administrative Commands

- AddUser(ユーザの追加)
- DeleteUser(ユーザの削除)
- AddRole(ロールの追加)
- DeleteRole(ロールの削除)
- AssignUser(ロールにユーザをアサインする)
- DeassignUser(ロールへのアサインを解除する)
- GrantPermission(ロールにパーミッションを割り当てる)
- RevokePermisiion(ロールへのパーミッション割り当てを解除する)

System Functions

- CreateSession（セッションの作成）
- DeleteSession（セッションの削除）
- AddActiveRole（アクティブロールの追加）
- DropActiveRole（アクティブロールの削除）
- CheckAccess（アクセス権限のチェック）

Review Functions

- AssignedUsers(ロールにアサインされたユーザの一覧を得る)
- AssignedRoles(ユーザにアサインされたロール一覧を得る)

Advanced Reviwe Functions

- RolePermissions(ロールに割り当てられたパーミッション一覧を得る)
- UserPermissions(ユーザにアサインされたパーミッション一覧を得る)
- SessionRoles(セッションにアサインされているアクティブロール一覧を得る)
- SessionPermissions(セッションにアサインされているパーミッション一覧を得る)
- RoleOperationsOnObject(あるロールが指定されたオブジェクトに対して許可されているオペレーション一覧を得る)
- UserOperationsOnObject(あるユーザが指定されたオブジェクトに対して許可されているオペレーション一覧を得る)

RBAC 仕様においては、オブジェクトの更新は考慮されていない。そこで本設計では、上記の API

に加えて、オブジェクトの更新を伴う WebDAV のために以下の API を追加した。

- AddObject（オブジェクト登録の追加）
- DeleteObject（オブジェクト登録の抹消）

3.2.2 WebDAV プロトコルの拡張

WebDAV クライアントと WebDAV サーバ間のプロトコルでは、RBAC 導入に伴い、セッションの概念を導入する必要がある。セッションに関連して追加が必要となるプロトコル拡張を示す。

以下に、各リクエストにおけるメッセージ作成方法およびレスポンスからの情報取得方法について述べる。

- セッション作成

セッション作成要求は、セッション要求専用コマンド RBAC を用いたリクエストを WebDAV サーバに送ることで行う。このとき、ユーザ情報を HTTP Basic ユーザ認証方式に従って指定する。

```
RBAC /dav/ HTTP/1.1
Host: localhost
Authentication: Basic xxxxxxxxxxxxxxxxxx
Content-Length: 0
```

ユーザが WebDAV サーバおよび RBAC サーバにより正しく認証されると、セッションが作成され、セッション ID がレスポンスに含まれて返送される。また、セッションが有効な間の全てのレスポンスのヘッダには、セッション ID とセッションのアクティブロールが示される。

```
HTTP/1.1 201 Created
RBAC-Session: xxxxxxxxxxxxxxxxxxxxxxxx
RBAC-Roles:
Content-Length: 0
```

- DAV リクエスト時のセッション指定

レスポンスで得られたセッション ID を記録し、次回以降のリクエストの送信時にヘッダに指定する。

```
OPTIONS /dav/ HTTP/1.1
Host: localhost
Authentication: Basic xxxxxxxxxxxxxxxxxx
RBAC-Session: xxxxxxxxxxxxxxxxxxxxxxxx
Content-Length: 0
```

- アクティブロール変更要求

有効なセッションへのアクティブロールの追加および削除は、RBAC 専用メソッドを用いて、ヘッダに追加削除したいロールを指定する。

```
RBAC /dav/ HTTP/1.1
Host: localhost
Authentication: Basic xxxxxxxxxxxxxxxxxx
RBAC-Session: xxxxxxxxxxxxxxxxxxxxxxxx
RBAC-Roles: +staff, +manager, -president
Content-Length: 0
```

- セッション破棄要求

セッション破棄を要求する場合は、RBAC 専用メソッドを使用して、破棄するセッションの ID をリクエストヘッダに指定する。

```
RBAC /dav/ HTTP/1.1
Host: localhost
Authentication: Basic xxxxxxxxxxxxxxxxx
RBAC-Session-Close: xxxxxxxxxxxxxxxxxxxxx
Content-Length: 0
```

3.3 アクセス権

アクセスチェックが必要な対象および権利は、リクエストの種類および対象リソースの状態によって異なる。

表 1 に、各メソッドと必要なアクセス権をまとめる。

本モジュールは、各 DAV 操作に応じて必要なアクセス権を判断し、RBAC 通信機能モジュールを介して RBAC サーバにアクセスチェックを要求する。

表 1 WebDAV メソッドとアクセス権の対応

WebDAV メソッド	確認するアクセス権
GET	対象に関する read 権
HEAD	対象に関する read 権
OPTIONS	対象に関する read 権
PUT（対象が存在する）	対象に関する write-content 権
PUT（対象が存在しない）	親コレクションに関する bind 権
PROPPATCH	対象に関する write-propaties 権
PROPFIND	対象に関する read 権
COPY（コピー先対象が存在する）	コピー元対象に関する read 権，コピー先対象に関する write-conten と write-propaties 権
COPY（コピー先対象が存在しない）	コピー元対象に関する read 権，親コレクションに関する bind 権
MOVE（移動先対象が存在しない）	移動元親コレクションに関する unbind 権，移動先親コレクションに関する bind 権
MOVE（移動先対象が存在する）	前述の権利に加えて移動先親コレクションに関する unbind 権
DELETE	親コレクションに関する unbind 権
LOCK（対象が存在する）	対象に関する write-content 権
LOCK（対象が存在しない）	親コレクションに関する bind 権
MKCOL	親コレクションに関する bind 権
UNLOCK	対象に関する unlock 権

4 ソフトウェア開発

前章で述べた設計に基づいてソフトウェア開発を行った。対象プラットフォームを表 2 に示す。

以下にそれぞれの機能モジュールの実装方法の概要について述べる。

4.1 RBAC サーバ

PostgreSQL の埋め込み SQL 機能を用いて RBAC データベースへのアクセスを実装した。また、クライアントからのリクエストメッセージを処理するための HTTP 通信処理、XML 解析処理を実装した。

4.2 WebDAV RBAC モジュール

Apache バージョン 2 の DSO モジュール機能およびフィルタ機能を用いて、WebDAV モジュールの前後でリクエストおよびレスポンスをフックする形の実装を行った。リクエストのフックでは、セッション作成およびアクセス権チェックを行い、レスポンスのフックでは、オブジェクトの更新情報を取得し、RBAC サーバに通知する機能を持つ。

4.3 WebDAV RBAC クライアント

既存の WebDAV クライアント cadaver を改造して、3.2.2節で述べたプロトコル拡張処理を実装した。また、Java を用いて GUI クライアントを実装した。

4.4 RBAC 管理クライアント

RBAC サーバと直接 HTTP 通信を行い、RBAC 管理コマンドのリクエストおよびレスポンスを処理する CUI クライアントを実装した。

表 2 動作環境

WebDAV サーバ環境	
マシンアーキテクチャ	Intel Pentium プロセッサ
OS	Linux 2.4 (TurboLinux 8 Server)
WebDAV **サーバ**	Apache 2.0.27 以降
RBAC サーバ環境	
マシンアーキテクチャ	Intel Pentium プロセッサ または Motrola PowerPC プロセッサ
OS	Windows 2000 以降 または MacOS 10.2 以降
Java	J2SE 1.4.2 以降
クライアント環境	
マシンアーキテクチャ	Intel Pentium プロセッサ または Motrola PowerPC プロセッサ
OS	Windows 2000 以降 または MacOS 10.2 以降
Java	J2SE 1.4.2 以降

5 まとめ

本稿では、RBAC 仕様に基づくアクセス制御を実現するための RBAC システム、および、RBAC システムを用いたセキュア WebDAV システムの設計・開発について述べた。RBAC は、組織構造と親和性が高いアクセス制御モデル、必要最小限のアクセス権行使、管理の容易性等の特徴により、大企業や政府等の大規模組織におけるアクセス制御に適している。WebDAV は、シンプルなプロトコル仕様、HTTP の自然な拡張による多種プラットフォームのサポート、ロックやバージョン管理等の分散協調オーサリングのための機能等、電子ドキュメントをはじめとするあらゆるリソースの分散協調アクセスの基盤プラットフォームとしてできしている。これらの特徴を統合したセキュア WebDAV システムは、以下のようなシステム構築に用いられることが想定される。

第 1 に、電子政府やコンソーシアム等の広範囲に分散する特定多数組織のユーザによる安全な情報共有およびオーサリングのシステムが考えられる。インターネットのようなオープンネットワーク環境を使ったシステム構築が行われる場合にも、WebDAV が持つ HTTP の特徴と RBAC の安全性を兼ね備える本システムにより、従来システムとの親和性、安全性、運営の容易性を両立することが可能となる。

第 2 の用途として、オープンソフトウェア開発に見られるような、地理的に離れた複数の開発者によるソフトウェアの共有ストレージ構築が考えられる。WebDAV システムによるバージョン管理により、ソフトウェア開発工程を適切に管理し、各モジュールのプログラマやベータテスタ等の役割に応じた適切なアクセス制御を RBAC で実現できる。

第 3 に、テレビ会議のような分散したユーザがインターネットを介して打ち合わせや議論を行う際の共有ホワイトボード構築の基盤として用いることを考える。HTML オーサリングツールや Wiki 等と組み合わせることで、インターネット上で会議出席メンバ専用の分散協調作業スペースを安全にかつ容易に構築できる。

6 参考文献

[1] RFC2518 "HTTP Extensions for Distributed Authoring -- WEBDAV" http://www.ietf.org/rfc/rfc2518.txt

[2] RFC3253 "Versioning Extensions to WebDAV (Web Distributed Authoring and Versioning)" http://www.ietf.org/rfc/rfc3253.txt

[3] RFC2617 "HTTP Authentication: Basic and Digest Access Authentication" http://www.ietf.org/rfc/rfc2617.txt

[4] RFC2616 “Hypertext Transfer Protocol -- HTTP/1.1” http://www.ietf.org/rfc/rfc2616.txt

[5] 情報処理推進機構, “WebDAV システムのセキュアな設定・運用に関する調査”, http://www.ipa.go.jp/security/fy14/reports/webdav/,

[6] Internet Draft “WebDAV Access Control Protocol” http://www.ietf.org/internet-drafts/draft-ietf-webdav-acl-13.txt