5W-8

ユーザによる設定を可能とする Proxy 型ネットワークアクセス制御方式の提案

関口 聖美† 黒羽 秀一†† 初谷 良輔†† 齋藤 孝道†
† 明治大学 †† 明治大学大学院

1 はじめに

XOOPS [1] や Wiki [2] など，Web 上で多数のユーザが，各々のファイルやディレクトリといったコンテンツを管理し，他のユーザに対して提供することが可能な Web アプリケーションが普及している．しかしながら，これらような Web アプリケーションでは，各ユーザが，所有するコンテンツに対してアクセス制御を行う場合，例えば Apache [3] の Basic 認証 [4] などを利用する．しかしながら，この方式では，アクセス制御の設定にサーバ管理者を必要とする．そのため，ユーザやコンテンツ数の増加，および，アクセス制御ポリシーの肥大化に伴い，サーバ管理者の負担が増大するだけでなく，コンテンツを提供したユーザ自身による独自のアクセス制御ポリシーの設定が困難である．

そこで，本論文では，複数のユーザから提供されたコンテンツを管理する Web アプリケーションにおいて，クライアントサイドからアクセス制御ポリシーを設定することができるプロキシ型のネットワークアクセス制御方式の提案と実装を示す．これにより，ユーザは所有するコンテンツ，すなわち，ファイルやディレクトリに対するアクセス権限の設定や他ユーザへの譲渡といった独自の運用ポリシーを設けることができる．

2 提案システム

2.1 概要

提案システムは，Internet Explorer や Netscape などの Web ブラウザであるクライアントと Apache を利用した Web サーバの間に配置されるリバースプロキシである．提案システムでは，Web ブラウザから送信された HTTP リクエストをもとに，Web サーバ上のコンテンツに対するアクセス権限を判定するモジュールを ACM (Access Control Module) と呼び，この ACM がその判定を行う際に参照するコンテンツに関するアクセス権限が記述されたリストを ACL (Access Control List) と呼ぶ．提案システムでは，ACL の実現に RDBMS(Relational Data Base Management System) を利用し，各ディレクトリごとにテーブルを用意する (2.3 副節)．また，クライアントサイドから，この ACL を編集するため，提案システムでは，ACL 編集用のユーザインタフェース (以降，EUI と呼ぶ) を提供している．

ここで，図 1 にクライアント，提案システム，Web サーバの配置関係を示す．図中の実線は，ACM を経由するクライアントと Web サーバ間の HTTP 通信を示し，破線は，クライアントと EUI 間の HTTP 通信を示している．

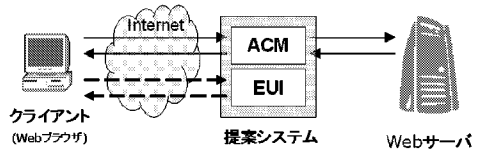

図 1: 提案システムの構成例

† Kiyomi SEKIGUCHI, Takamichi SAITO
†† Shuichi KUROBA, Ryosuke HATSUGAI
{skgchi, kuroba, hatsugai, saito}@cs.meiji.ac.jp
Meiji University(†), Graduate School of Meiji University(††)
1-1-1 Higashimita, Tama-ku, Kawasaki-shi, Kanagawa, 214-8571, Japan(†)(††)

2.2 実装環境

ACM は，Fedora Core 4 (kernel 2.6.11) 上に，JDK (JAVA 2 SDK，Standard Edition，v 1.4.2_12) と RDBMS のひとつである MySQL 5.0.22 [5] を用いて実現した．次に，ACM が管理する ACL (MySQL) にアクセスし，その編集をクライアントサイドから行うための EUI は，JSP (Java Server Pages) によって実装し，そのサーブレットコンテナと Web サーバには，Tomcat 5.0.28 [6] と Apache 2.2.2 をそれぞれ利用して，実現した．

2.3 提案システムの詳細

2.3.1 ACL の構造

提案システムの ACL は，図 1 の Web サーバ上でユーザごとに管理されるファイルおよびディレクトリの階層と同じようにツリー化した MySQL のテーブルの集合である．例えば，Web サーバ内に図 2 に示すようなファイル，および，ディレクトリが存在する場合，ACL は図 3 のようになる*．下記に，ACL のテーブルが保持する各フィールドの役割を示す．ただし，以降，図 3 に示すファイルやディレクトリの所有者は，ユーザ Alice とし，Alice が作成したアクセス制御ポリシーを適用するユーザとして，Bob，Carol，Eve が存在するという前提で，説明する:

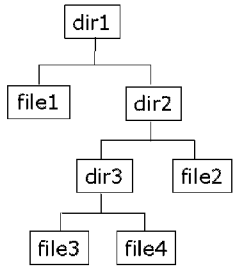

図 2: ファイルおよびディレクトリ階層例

table: top_dir

file	allow	deny	delegate	owner
dir1	All:rw			Alice

table: dir1

file	allow	deny	delegate	owner
file1	All:rw	Carol:rw		Alice
dir2	All:rw		Bob:O	Alice

table: dir2

file	allow	deny	delegate	owner
dir3	All:rw			Alice
file2	All:rw			Alice

table: dir3

file	allow	deny	delegate	owner
file3	All:rw			Alice
file4	All:rw			Alice

図 3: 提案システムの ACL 例

file: ファイル名または，ディレクトリ名を記述する．
allow: file フィールドに記述されたコンテンツに対してアクセスを許可するユーザを記述する．アク

* テーブル名がディレクトリ名に対応している．また，top_dir はルートディレクトリのアクセス制御ポリシーを設定するためのテーブルである．

セス制御は,「読み込み」と「書き込み」の 2 つがあり，それぞれ r と w のフラグ表記で指定する．例えば,「Carol:r-」と記述した場合，Carol は「読み込み」のみが許可される．

deny: file フィールドに記述されたコンテンツに対してアクセスを拒否するユーザを記述する．上述の allow フィールドの場合と同様，このフィールドも r と w のフラグ表記を用いる．例えば「Carol:-w」と記述した場合，Carol は「書き込み」のみが拒否される．

delegate: file フィールドに記述されたコンテンツに対するアクセス制御ポリシーを設定するための権限を他のユーザに委譲する場合に利用するフィールドであり，ここには権限を委譲するユーザ名を記述する．委譲の条件として,「削除，追記可能」と「追記のみ可能」があり，フラグ O とフラグ A でそれぞれ表記でする．O 権限が与えられたユーザが可能な操作は，設定済みのアクセス制御ポリシーの削除，追記，アクセス制御の権限の再委譲，コンテンツの新規作成の 4 つである．また，A 権限が与えられたユーザが可能な操作は，設定済みのアクセス制御ポリシーの追記，アクセス制御の権限の再委譲，コンテンツの新規作成の 3 つである．

owner: file フィールドに記述されたコンテンツの所有者を記述する．これは，このコンテンツを委譲されたユーザが，その所有権を奪取し，本来の所有者がそのコンテンツに対するアクセス制御ポリシーの変更ができなくなることを防ぐために利用する．そのため，一度設定されたこのフィールドの記述を変更することはできない．

各フィールドに記述するユーザ名が複数になる場合は，カンマで区切る．また，allow と deny フィールドには，すべてのユーザが対象であることを示す All という表記を利用することもでき，これにより，各フィールドの記述数を減らすことができる．さらに，allow と deny フィールドのフラグ表記のうち,「-」と表記されている部分は，フラグが適用されないことを示す．

2.3.2 ACL の設定方法

EUI を利用したファイルとディレクトリに対するアクセス制御ポリシーの設定方法を下記に示す:

(1) ファイルに対するアクセス制御ポリシーの設定

例えば，図 3 に示す ACL のアクセス制御ポリシーを設定した Alice が，新たにユーザ Carol の file2 に対する読み書きを拒否するためのアクセス制御を設定する場合，Alice は，図 3 の table:dir2 を図 4 のように変更する．

table: dir2

file	allow	deny	delegate	owner
dir3	All:rw			Alice
file2	All:rw	Carol:rw		Alice

図 4: ポリシー設定後の ACL のテーブル例 1

(2) ディレクトリに対するアクセス制御ポリシーの設定

Alice が，dir3 の配下にある file3 と file4 に対する書き込みを Carol に対して拒否し，さらに，file4 に対しては読み込みも拒否する場合，Alice は，図 3 の table:dir2 と table:dir3 を，図 5 に示すようにそれぞれ変更する．

2.3.3 ACM によるアクセス制御

ACM の処理手順について説明する．ここでは例として，図 2 のディレクトリ構造とそれに対応する図 3 の ACL が，Alice によって設定されており，この状況下で Carol が file1 にアクセスを試みた場合の ACM の処理を説明する:

(1) Carol は，Web ブラウザを利用して ACM 経由で Web サーバにアクセスする．このとき，Web ブラウザから送信される HTTP リクエストには，file1 までのファイルパス (/dir1/file1) が含まれている．

table: dir2

file	allow	deny	delegate	owner
dir3	All:rw	Carol:-w		Alice
file2	All:rw	Carol:rw		Alice

table: dir3

file	allow	deny	delegate	owner
file3	All:rw	Carol:-w		Alice
file4	All:rw	Carol:rw		Alice

図 5: ポリシー設定後の ACL のテーブル例 2

(2) ACM は，先頭ディレクトリ dir1 に対応する ACL のテーブルを参照し，allow と deny フィールド (図 3 の①) を確認する．ここでは，allow フィールドに All と記述されており，deny フィールドにはユーザが明記されていないため，ACM は，Carol は dir1 へのアクセス (読み書き) は許可されていると判断する．同様に，ACM は，ファイルパスの次のリソースに対応する ACL のテーブルを確認する．ここで，図 3 の② の deny フィールドに「Carol:rw」とあるため，ACM は，Carol は file1 に対するアクセスは許可されていないと判断する．

(3) ACM は，Carol に対して，アクセスの拒否を伝える通報を送信する．

2.3.4 アクセス制御の権限委譲

アクセス制御の権限委譲について例を用いて示す．

例えば，Alice が，Bob に dir2 以下の全てのファイルとディレクトリに対するアクセス権限の変更を許可する場合，Alice は図 3 の③のように delegate フィールドを設定する．ここで，Alice は，O 権限を Bob に与えているため，Bob は，dir2 以下のコンテンツに対するアクセス権限の追記，削除，再委譲，また，コンテンツの新規作成が可能となる．ただし，Bob が dir2 以下に作成したコンテンツに対しては，このディレクトリの本来の所有者でもある Alice もアクセス制御ポリシーを設定することができる．

さらに，例えば，Bob が，dir3 以下のコンテンツを Eve に再委譲するには，Bob は，図 3 の④の delegate フィールドに「Eve」と記述する．その際，Bob が，Eve に付与することができるフラグは，Bob が Alice から受け取った権限に依存する．Bob が，O 権限を Alice から付与されている場合，Eve には O 権限，または，A 権限を付与することができるが，Bob が A 権限を付与されている場合には，A 権限のみを Eve に付与することができる．提案システムでは，同様の手順にて，コンテンツの権限委譲を繰り返すことができる．

3 まとめ

本論文では，ユーザによるコンテンツ管理を行う Web アプリケーションに対してクライアントサイドからアクセス制御ポリシーの設定を行うためのフレームワークとそれを利用したプロキシ型のアクセス制御方式を提案した．また，Web アプリケーション上のコンテンツをディレクトリ単位で他のユーザに委譲する方法を提案した．

今後の課題として，多数の利用者が存在する環境下での提案システムのパフォーマンス計測があり，これにより，提案システムのボトルネックを特定する．また，複数のバックエンド Web サーバが存在する環境下でも提案方式を適用できるように実装修正を行う．

参考文献

[1] http://xoopscube.org/
[2] http://pukiwiki.sourceforge.jp/
[3] http://www.apache.org/
[4] http://httpd.apache.org/docs/1.3/howto/auth.html
[5] http://www.mysql.com/
[6] http://tomcat.apache.org/