ユーザ認証としての全学 ID システムの習得

藤原冨未治*、若松　進*
*工学研究科・工学部技術部 電子・情報技術系

はじめに

名古屋大学では、各種事務書類を作成するにあたり Web を用いた全学事務案内システム（以下、事務システム）を利用している。現在事務システムを利用する時の認証方法として、職員認証データベース（以下、職員認証 DB）を使用し、職員個々に発行されている ID（職員 ID）で個人を特定している。この職員 ID は、工学研究科で運用中の複数のシステム（会議資料参照システム等）の認証用に利用されている。しかし職員認証 DB 用サーバの運用上、近い将来廃止が予定されており、現在運用中のシステムを継続して利用するためには、ユーザ認証を他の認証方式に変更する必要性が生じた。

名古屋大学では、現在職員、学生に対して学内統一用に全学 ID の発行を開始して、学内での情報サービスでの二次利用の推進も行っているため、開発中のシステムも含め認証部分を職員 ID から全学 ID への切り替え作業の準備中である。

このため、全学 ID の認証メカニズムとその利用方法の習得が重要課題となっている。

１．現認証の問題

現行の認証方式ではシステムサーバから認証のために職員認証 DB にアクセスし、職員 ID の問い合わせを行う際、インターネット上を流れるデータが暗号化されておらず盗聴される恐れがあることが以前より問題視されていた(図 1)。フレキシブルな運用を行うために DB 上ではパッケージ化されたスクリプトが用意されているが、情報が盗聴された場合非常に問題があること、今後 DB サーバの運用が廃止の方向であることなどの理由で、全学 ID を用いた CAS 認証を利用する必要性が生じた。

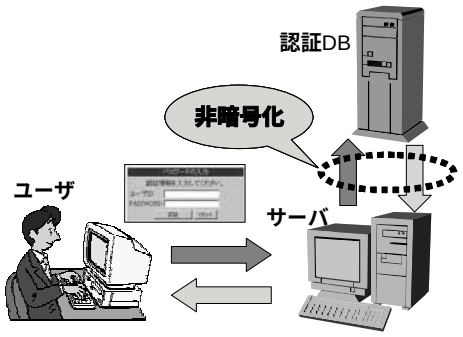

図１　現認証での問題点

２．CAS とは

CAS は Yale 大学で開発された認証機構であり Central Authentication Service の頭文字を取ったものであり[1]、Web ベースのアプリケーションに対してシングルサインオン環境を実現できる。

特徴として

1. HTTP リダイレクション、Ticket Granting Cookie（TGC）、Service Ticket（ST）[1]という一般的な Web 技術を用いているため設定が容易である。
2. CAS 認証を利用するには CAS サーバに問い合わせるための CAS 認証用のライブラリをシステ

[1] URL パラメータとしてセットされる

ム側で用意する必要がある。ライブラリには Java、PHP、Perl、PL/SQL 等が用意されている。

3.認証のためのユーザ ID とパスワードは暗号化された上で CAS サーバにのみ送信されるため、通信路において情報漏洩の心配がない。

このように CAS 認証では現在使用している職員認証の問題点である認証データの暗号化が施されており、全学用に運用が開始されているためその安定性も保証されている。

3．CAS を用いたユーザ認証

ここでは､実際にシステム側にログイン認証用のサーブレットを置き CAS 認証で本人認証を確認後、認証属性からそれに応じた値を引き出せるかの検証をした。

1）認証準備

システム側の開発言語に Java を用いているため､CAS サーバとの通信のやり取り用のライブラリとして Yale 大学から供給されている Java 用 CAS クライアントパッケージを用いた。パッケージ自体は Java 実行環境下である~/WEB-INF/lib 以下に配置し、Java サーブレットからアクセスできるように設定した。またサーブレットコンパイル用にクラスパスにも追加した。次にテスト用のユーザ認証サーブレットをサーブレット実行環境である~/WEB-INF/classes/以下に配置し、コンパイルして class ファイルを作成した。

2）認証テストの流れ

今回の認証テストの大まかな流れは、以下の通りである。

①クライアントから Web ブラウザでシステムにアクセスすると CAS サーバにリダイレクトされ CAS サーバ側のログイン画面が表示される(図２)。

②ログイン ID とパスワードで CAS 認証し、本人を確認する(図３)。

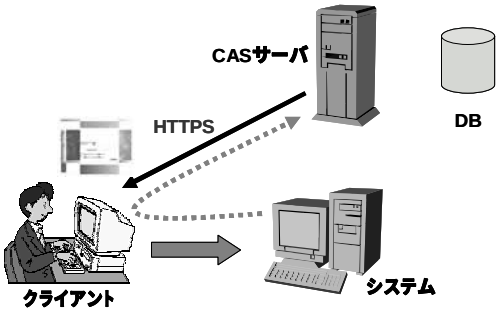

図２　認証動作（１）

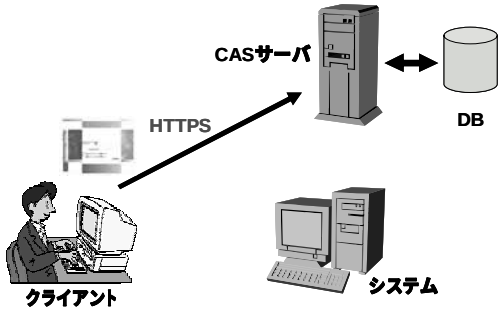

図３　認証動作（２）

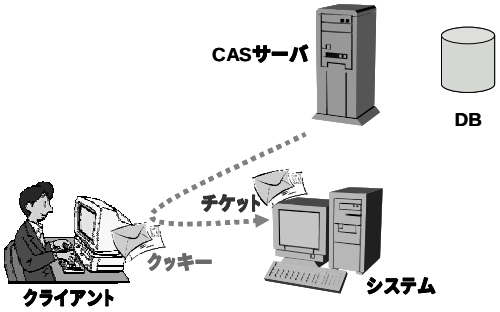

図４　認証動作（３）

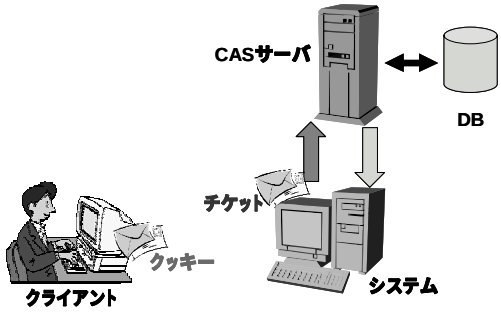

図５　認証動作（４）

③CAS サーバ側で発行したチケットを付加した値がシステム側にリダイレクトされる(図４)。
④送られて来たチケットを用いてシステムから CAS サーバに問い合わせを行い、認証を再検証する(図５)。
⑤得られた個人認証から属性情報を取り出し、各種値を表示する(図６、図７)。

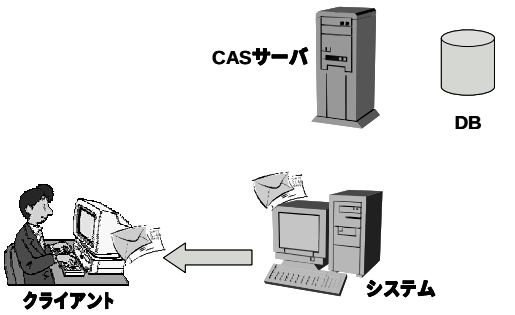

図６　認証動作（５）

図７　認証属性の表示例

３）認証テスト

テスト用サーブレットとして、CAS クライアントパッケージを組み込んだプログラムをシステム側で用意し(図８)、CAS サーバ側ではアクセス許可の設定を施してもらった。

実行結果として

a) システムにアクセスするとリダイレクトされサーバのログイン画面が表示された。
b) ID とパスワード入力し、CAS 認証で個人が認証されることを確認した。
c) 認証後、属性情報から氏名を取得しようとするとエラーとなった。
d) 氏名情報である漢字コードで文字化けがおこった。

```
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import edu.yale.its.tp.cas.client.*;

public class CasLoginServlet extends HttpServlet {

    private static final String CAS_SERVICE_URL = "http://xx.xxx.xxxx.nagoya-
u.ac.jp/xxxxx/servlet/CasLoginServlet";

protected void perform(HttpServletRequest request, HttpServletResponse
respo
nse ) throws ServletException, IOException {

    String userid = null;
    String fullName = null;
        略

CasClient cas = new CasClient() ;
    cas.setServiceURL(CAS_SERVICE_URL) ;
    if (!cas.casPerform(request, response)) return ;
    Map r = cas.getResult() ;
        略
```
図８　テスト用サーブレット

c)に関しては認証後サービスチケットが URL に埋め込まれてシステムにリダイレクトされる過程でチケット値が埋め込まれていなかったため、システムと CAS サーバとの再検証が行われておらず、属性情報も取得出来ない状態であった。これは CAS 認証が名古屋大学用にカスタマイズされているためで通常の CAS クライアントパッケージを使用しただけでは、取得できない変数が存在したためと判明した。

d)の問題では、CAS がデフォルトの文字コードとして UTF-8 を指定しているのに対してシステム側の文字コードは EUC コードであったため、この問題が起こったことが判明した。

上記の問題解決のために CAS クライアントパッケージでエンコードする文字コードを UTF-8 にするように改良し、名古屋大学用にカスタマイズした CAS クライアントパッケージを作成、配布し

てもらうことで解決した。

c)、d)の問題解決により CAS 認証での認証後、個人属性情報である ID と氏名を取得し表示することが出来た。

４．まとめ

本研鑽では、CAS システムの概念から CAS 認証までの一連の動作を含めた基本的な知識を習得することが出来た。

Java サーブレットで作成した CAS 認証用サンプルコードと名古屋大学用にカスタマイズされた CAS クライアントパッケージを用いることで、個人認証がシステムサーバ上で正しく動作することを確認した。

また、CAS 認証後にアクセスしたユーザの属性情報である ID と氏名が取り出せることが確認できた。これにより現在開発中のシステムも含め CAS 認証の実装方法を確立することができた。

５．今後の予定

今回の研鑽では、CAS 認証を利用して個人認証が正しく行われるかどうかのテストに限定し、取得する属性情報も個人 ID と氏名の情報の取得だけとした。今後我々が開発してきたシステムに応用するためには CAS 認証後に取得できる属性情報についての知識を広げることと、各々のシステムに応じた属性情報を引き出すための検証を行う必要がある。

その後今まで開発して来たシステムの認証部分を職員 ID 認証から CAS 認証（平成 18 年 4 月から運用が開始される名大 ID）への書き替え作業を順次行っていく予定である。

謝辞

本研鑽は、以前よりプログラム開発を共同で行っている全学技術センター情報通信技術系の太田芳博技術専門員の協力を得て実施した。

参考文献

[1] Central Authentication Service
http://www.ja-sig.org/wiki/display/CAS/Home

[2] ESUP-Portail: open source Single Sign-On with CAS(EUNIS2004)
http://perso.univ-rennes1.fr/pascal.aubry/presentations/cas-eunis2004/

[3] Single Sign-On open-source avec CAS(Central Authentication Service)
http://www.esup-portail.org/consortium/espace/SSO_1B/cas/jres/cas-jres2003-article-web.htm

[4] 梶田将司,内藤久資,小尻智子,平野靖,間瀬健二, CAS によるセキュアな全学認証基盤による名古屋大学ポータルの運用, 第３回 WebCT Conference 予稿集, pp. 115-120(2005).

[5] 梶田将司,内藤久資,小尻智子,平野靖,間瀬健二, CAS によるセキュアな全学認証基盤の構築, 名古屋大学情報連携基盤センターニュース, Vol.4, No.3, 2005.8, pp. 179-187

[6] 内藤久資, 梶田将司, Central Authentication and Authorization Service －Web Application のための新しい認証システムの試み－, 京都大学数理解析講究録, 1446, 「電子情報交換に関する最近の話題」, 14-39, (2005)