平成 13 年度

学士学位論文

遠隔コンテンツ操作方式

A Form Operation For Remote Contents

1020321　　福留　一夫

指導教員　　清水　明宏

2002 年 2 月 8 日

高知工科大学 情報システム工学科

要 旨

遠隔コンテンツ操作方式

福留　一夫

近年，インターネットが普及しサーバ/クライアントシステムの導入されたイントラネット内のファイルサーバへ，イントラネット外にある端末からアクセスしファイルを送受信したいという要求が高まっている．ファイルサーバにアクセスする方法として，FTP(File Transfer Protocol) が考えられるが，イントラネット内の重要ファイルや個人ファイルの流出を防ぐためファイアウォールを設置し FTP のサービスは提供されていない場合が多い．本論文では，通信に HTTP(HyperText Transfer Protocol) を用いて出先端末の環境に依存しないファイル送受信サービスの提案する．さらに，通信パケットを暗号化し，ユーザ認証に SAS 認証方式を用いた，より安全サービスと試作システムについて述べる．

キーワード　ファイル送受信，FTP，認証，HTTP，SAS

Abstract

A Form Operation For Remote Contents

Kazuo Fukudome

Recently, the internet was spread. The demand of wanting to access from the terminal outside the intranet to the file server in the intranet which was introduced server/client system for receiving the file service is increasing. FTP (File Transfer Protocol) is one of the file service that a file server accessed from a terminal. But FTP is not offered in many cases to prevent going out important and personal files in intranet.In this section, I propose the File service using HTTP (HyperText Transfer Protocol) between server and client for breaking firewall and is independent of the Client. And this service encrypt the HTTP packet, and using the sas system make the file service more secured. And I refer to a trial system.

key words File service, FTP, HTTP, SAS

目次

図目次

第 1 章

はじめに

近年，コンピュータ技術の進歩により，小型で安価パソコンが出回るようになった．また，高速で安価なインターネット接続サービスが提供され，家庭でもインターネットが利用できるようになった．さらに，インターネットカフェ，駅構内の無線 LAN サービス，ホテルや飲食店などでのインターネット接続サービスなどが広まり，公共の場所でもインターネットが使用できるようになりつつある．

企業や教育機関でも，インターネットの接続はなされ，さらにサーバ/クライアントシステムによって，独自にイントラネットを形成し，重要ファイルや個人のファイルをファイルサーバで保存するようになった．しかし，出張先や会議場といった出先で，ファイルサーバに蓄積されている情報が必要になる場合がある．そうしたことから，出先でファイルサーバからのファイル送受信サービスを受けたいという要求が高まってきている．しかし，イントラネットでは外部に情報が漏洩しないようファイアウォールを設置し，不特定な IP アドレスのアクセスを拒否している．よって，出先からファイルサーバへのアクセスはできず，ファイル送受信サービスを受けることはできない．

そこで本論文では，このような場合でもファイル送受信できる”遠隔コンテンツ操作方式”を提案し，またこれをもとに構築した試作システムの概要について述べる．

第 2 章

研究背景

第 1 章でも述べたが，出先からイントラネット内のファイルサーバへアクセスし，ファイルの送受信サービスを受けたいという要求が高まっている．ファイル送受信サービスの 1 つに，FTP(File Transfer Protocol) が考えられる．しかし，企業や教育機関のイントラネットでは，重要ファイルや個人ファイルの流出を防ぐためファイアウォールを設置し，FTP のサービスは提供されていない場合が多い．また，認証サーバを経由したダイヤルアップ IP 接続を用いコールバックを利用したユーザ認証もあるが，この場合あらかじめサーバに電話番号を登録する必要がある．また，出先端末にもダイアルアップの設定を施さなければならないため，あまり実用的ではない．他にも VPN(Virtual Private Network) や SSH(Secure SHell) を導入し，ネットワーク上で流れるデータを制御したり，暗号化してファイル送受信を安全に行う方法もある。しかし，これらの方法でも先のダイアルアップ IP 接続と同様に，出先端末の環境を変更する必要がある．

2.1 Pop-up Mail

端末環境を変更することなく内部ネットワークのメールサーバから，中継サーバを経由し電子メールを送受信することができる Pop-up Mail サービス (図 2.1) がある．このサービス方式では，電子メールの転送を内部ネットワークのメールサーバと，インターネットに接続された中継サーバと，出先の端末の 3 者間で行うサービス方式である．3 者間の転送プロトコルには，HTTP(HyperText Transfer Protocol),SMTP(Simple Mail Transfer

Protocol) を利用し，ユーザ認証にワンタイム認証方式である PERM(Privacy Enhanced Information Reading and Writing Management Protcol) 認証を用いている．このサービスではインターネットに接続している端末の Web ブラウザから，中継サーバへアクセスするだけで外部から電子メールを送受信することが可能である．また，昨年度の本研究室の研究によって PERM 認証から SAS 認証を用いることで，第 3 者によるなるりすまし行為の可能性が減る，サーバの保有するデータ量が減るといったことが証明されており，効率の良いサービスが可能になることがわかっている．しかし本研究の目的とするファイル送受信のサービスは行うことができない．

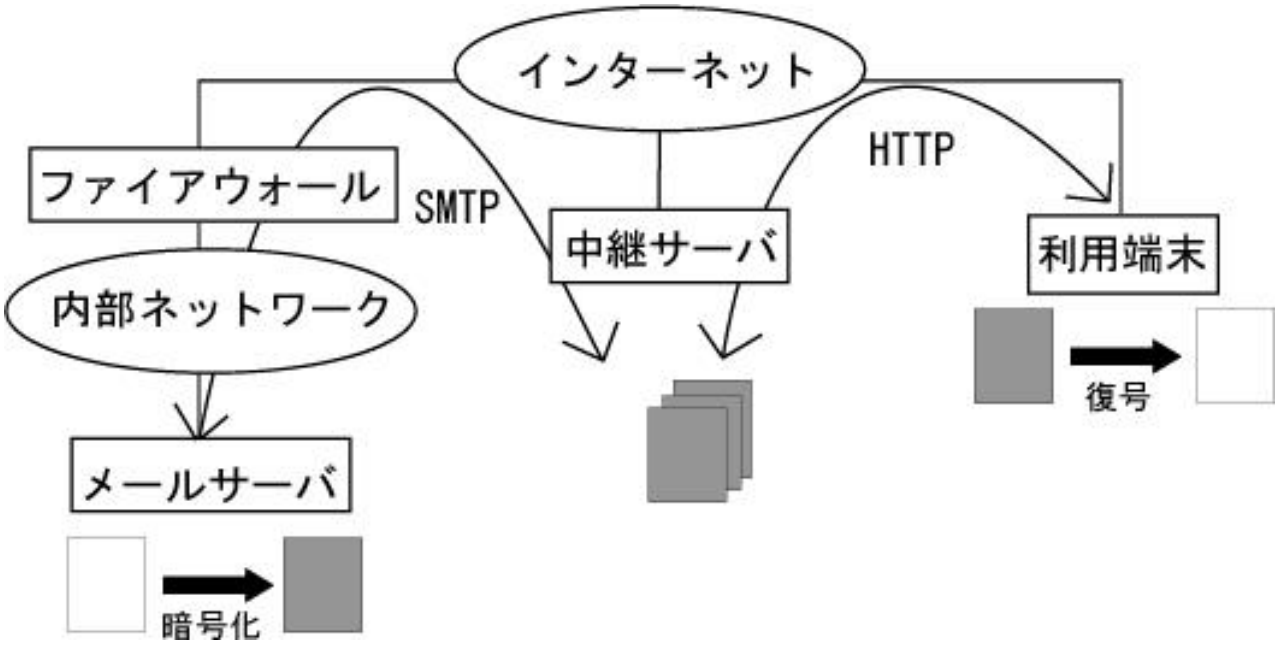

図 2.1　Pop-up Mail サービス

2.2　3 者間ファイル転送方式

Pop-up Mail サービスをもとにした，3 者間によるファイル転送方式 (図 2.2) が本研究室より提案された．

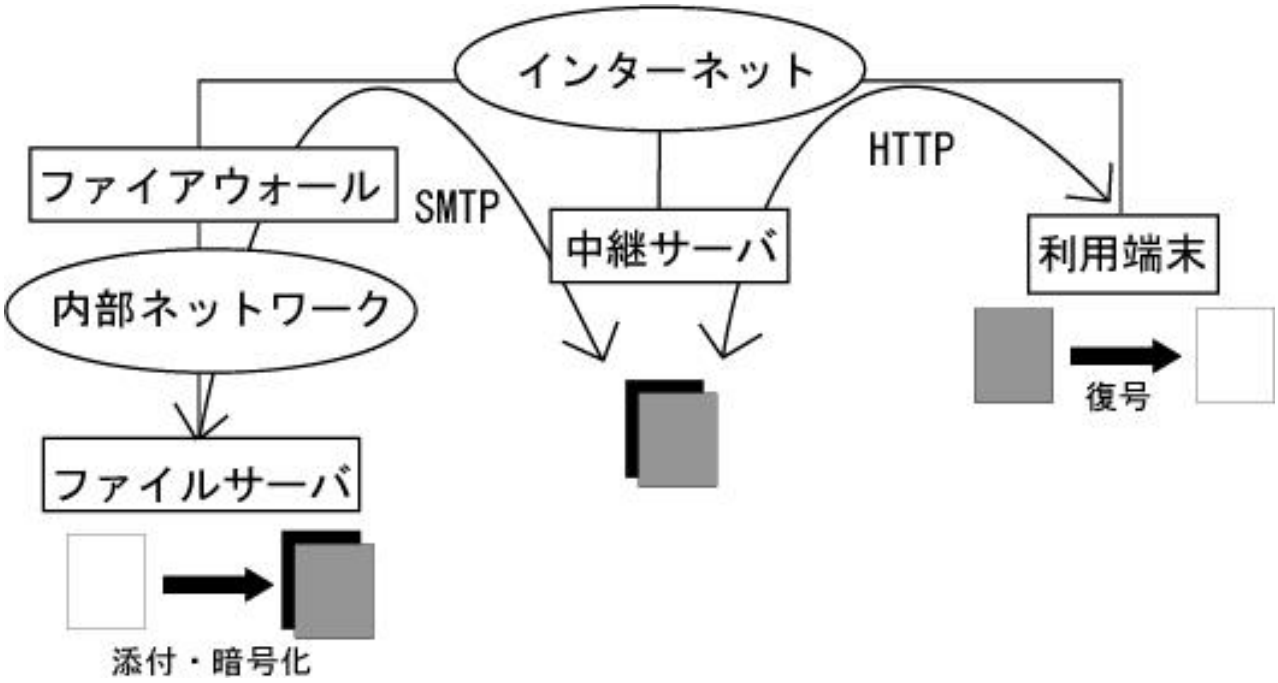

図 2.2　3 者間ファイル転送方式

3 者間ファイル転送方式では，ファイルサーバにあるファイルを電子メールに添付し，電子メールを暗号化し，中継サーバに転送する．その後，出先端末からファイルを DownLoad し復号することでファイル受信が可能になる．ファイルサーバと中継サーバ間の通信には SMTP を使い，中継サーバと出先端末間の通信には HTTP を使う，そしてユーザ認証には SAS 認証を用いる．しかし，SMTP を用いたファイル転送には，通信品質に問題があり実用的でない．

本研究では，この 3 者間ファイル転送方式をもとに出先端末の環境変更を行わずに，またファイルを安全に送受信する方式として，3 章より述べる遠隔コンテンツ操作方式を提案する．

第 3 章

遠隔コンテンツ操作方式

遠隔コンテンツ操作方式では，イントラネットに設置されたファイアウォールを回避し，出先端末の環境に依存しないサービスを提供するため，ファイルサーバと出先端末間の通信に HTTP を用いる．また，第 3 者による盗聴や成りすましを防ぐため，ユーザ認証を行い，HTTP のパケット暗号化する．ユーザ認証には，毎回認証情報が変わるワンタイムパスワード方式である SAS 認証方式を採用し，SAS 認証データに HTTP のパケットを含ませることで，暗号化を行う．これらの点をふまえ，そして 3 者間ファイル転送方式をもとにした遠隔コンテンツ操作プロトコルとして，次の 2 つのプロトコルを提案する．

3.1　遠隔コンテンツ操作プロトコル 1

1 つ目のプロトコルは，3 者間ファイル転送をもとにしたプロトコルである．これを図 3.1 に表す.

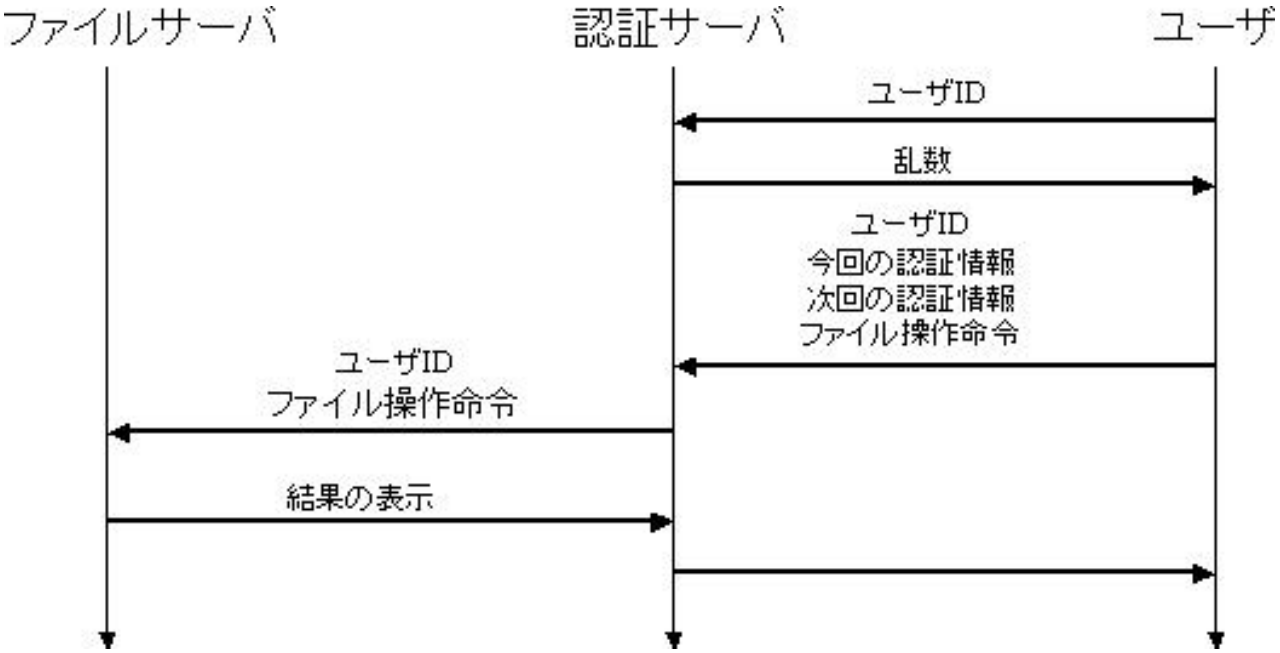

図 3.1　認証サーバを用いたプロトコル

このプロトコルでは，最初にユーザは認証サーバへリクエストを送信する．その後サーバは，ユーザに今回の認証情報の作成に必要な前回の乱数を返す．ユーザは，前回の乱数から今回の認証情報，また新たな乱数を生成し次回の認証情報を作成する．そして今回の認証情報，次回の認証情報，ファイル操作命令を認証サーバに送信する．認証サーバでは，認証情報から正当なユーザと判断すると，ファイルサーバへファイル操作命令が送信される．ファイルサーバでは，受信したファイル操作命令を実行しその後，結果の HP を作成する．ユーザは，その HP にアクセスすることで結果を参照することができる．このプロトコルの利点として，認証サーバを設置することで，ファイルサーバに直接アクセスすることが出来ないのでセキュリティ面で安全だが，2 つのサーバを管理しなければならない問題が発生する．また，認証サーバとファイルサーバ間の通信も不完全である．

3.2　遠隔コンテンツ操作プロトコル 2

2 つ目のプロトコルは，1 つ目のプロトコルより認証サーバを除いたものである．これを図 3.2 に表す．

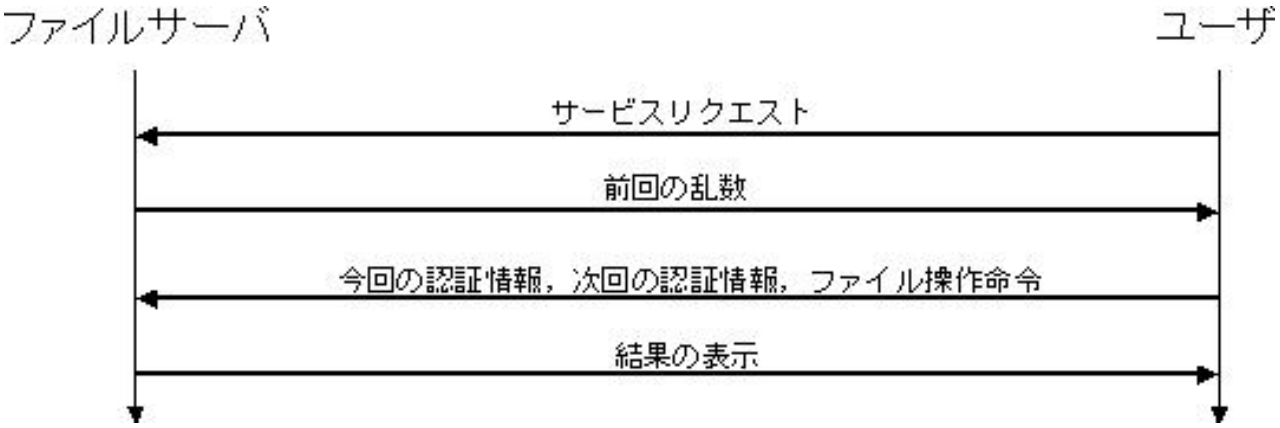

図 3.2　認証サーバを用いないプロトコル

最初にユーザは出先端末から，ファイルサーバへリクエストを送信する．その後ファイルサーバは，ユーザに今回の認証情報の作成に必要な前回の乱数を返す．ユーザは，前回の乱数から今回の認証情報，また新たな乱数を生成し次回の認証情報を作成する．そして今回の認証情報，次回の認証情報，ファイル操作命令をファイルサーバに送信する．ファイルサーバでは，認証情報から正当なユーザと判断すると，ファイル操作命令を実行し，結果の HP を作成する．ユーザは，その HP にアクセスすることで結果を参照することができる．このプロトコルの利点として，管理するサーバ数が 1 つになるが，直接ファイルサーバにアクセスされるので，何らかの被害が発生した場合，ファイルサーバが直接被害をこうむる．しかし，4.1 章で述べる ApacheJServ 機能を用いることで，この問題を解決できる．

第 4 章

試作システム

試作システムは，3.2 章での遠隔コンテンツ操作プロトコル 2 をもとに構築した．試作システムのプログラムには，出先端末環境に依存しない動的な HP を作成するため JavaApplet と JavaServlet を使用している．

4.1 試作システム環境

試作システム環境は，研究室内のネットワークを利用して構築した．まず，ファイルサーバに PentiumII333MHz，メモリ 128M，OS に VineLinux2.1.5 をインストールしたマシンを準備し，WWW サーバに Apache1.3.19 を用いた．ServletEngine には，代表的なものに TomCat,JRun,ApacheJServ があるが，今回は，ApacheJServ1.1.2 を採用した．ApacheJServ を採用した理由に他の ServletEngine に見られない，次の 2 点の特徴があるからである．

1. ApacheJServ は，Apache と ApacheJServ 間でパスワードを決めて通信するので，アクセスされる Apache を指定できる．
2. ApacheJServ は，Zone という設定ファイルで JavaServlet の個人環境を設定することが可能である．

1 点目より，他のサーバを媒介しての攻撃を防げると考えられる．また，2 点目よりファイルサーバのユーザアカウントごとに ZONE と呼ばれる定義ファイルを作成することで，アクセスする JavaServlet のディレクトリによって，ユーザを断定することが可能である．このことより，ユーザのパーミッションを対応させることが可能になり，ファイルサーバの

4.1　試作システム環境

ファイルシステムで，ファイル管理を行うことができる．これらの点から，本システムに最適な ServletEngine であるいえる．また，クライアントマシンとして Pentium100MHz，メモリ 64M，OS に Windows95 をインストールしたマシンを準備し，Web ブラウザには，InternetExplorer5.5SP2 を用いた．今回の試作システムの実験環境を，図 4.1 に表す.

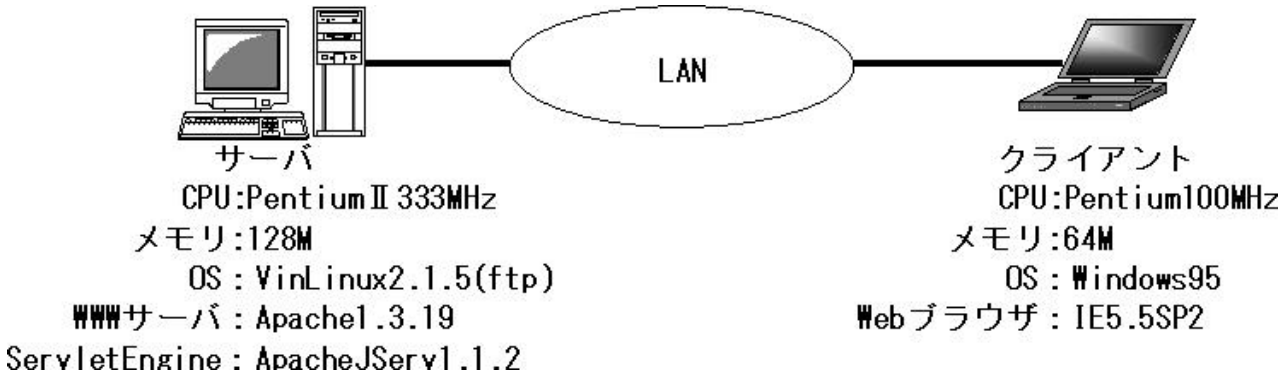

図 4.1　試作システム環境

4.2　出先端末の仕様

4.2.1　認証画面

ユーザは，ファイルサーバへブラウザを用いてアクセスすることを前提にしている．よって，試作システムでは図 4.2 のようなログインフォームで ID と PASSWORD を入力しファイルサーバへ送信する．ファイルサーバでは、ユーザ認証として SAS 認証を行い安全性を高めている．

図 4.2　ログインフォーム

4.2.2　SAS 認証

SAS 認証による認証手順は、下記に示す.

記号の定義

A:UserID

S:Password

n:認証の回数を示す 0 以上の整数

N_n:n 回目の認証で使用する乱数

E_n^1:N_n を用いて 1 回，一方向性関数をかけたデータ

E_n^2:N_n を用いて 2 回，一方向性関数をかけたデータ

$\oplus$:排他的論理和 XOR

前処理 (n=0)

1. ユーザは，$E_0^1 \leftarrow$ E (A，S $\oplus$ N_0)

   $E_0^2 \leftarrow$ E (A，E_0^1) を計算する.
2. ユーザは，A，E_0^2 を安全なルートでファイルサーバへ送信する.
3. ファイルサーバでは A，E_0^2 を保存する.

認証処理 (n=k)

1. ユーザは，ファイルサーバへサービス要求を送信する.
2. 要求を受け取ったサーバは，ユーザに前回の乱数 N_{k-1} を返す.
3. ユーザは乱数 N_{k-1} から今回の認証情報を作成し，

   $E_{k-1}^1 \leftarrow$ E (A, S $\oplus$ N_{k-1})

   $E_{k-1}^2 \leftarrow$ E (A, E_{k-1}^1)

また乱数 N_k を生成し次回の認証情報を作成する．

$E_k^1 \leftarrow \mathrm{E}\ (\mathrm{A},\ \mathrm{S} \oplus N_k)$

$E_k^2 \leftarrow \mathrm{E}\ (\mathrm{A},\ E_k^1)$

4. ユーザは，A，N_k，$E_{k-1}^1 \oplus E_{k-1}^2$，$E_k^2 \oplus E_{k-1}^2$ をファイルサーバへ送信する．
5. サーバでは，受け取った $E_{k-1}^1 \oplus E_{k-1}^2$ に，サーバに登録されていた E_{k-1}^2 を用いて，$E_{k-1}^1 \oplus E_{k-1}^2 \oplus E_{k-1}^2$ を行い E_{k-1}^1 を取り出す．

   取り出した E_{k-1}^1 に，

   $E_{k-1}^2 \leftarrow \mathrm{E}\ (\mathrm{A},\ E_{k-1}^1)$

   を行い，サーバに登録されいた E_{k-1}^2 と比較する．
6. 正当な認証情報であれば，$E_k^2 \oplus E_{k-1}^2 \oplus E_{k-1}^2$ を行い取り出した E_k^2 を次回認証情報としてサーバに登録する．

4.2.3　メイン画面

ユーザは、SAS 認証が成立すると，図 4.3 で示すメイン画面へと画面が変わる．

図 4.3　メイン画面

メイン画面では，画面上半分をコマンド送信フォームとして表示し，ユーザはファイル操作命令をこのフォームから，ファイルサーバへ送信する．画面下半分には，ファイルサーバからの結果を表示している．今回のシステムでは，ユーザは結果を見ながらコマンドを送信する必要があるためこのようなインタフェースを設計した．

4.2.4　コマンド送信フォーム

コマンド送信フォームを拡大したものが図 4.4 である．

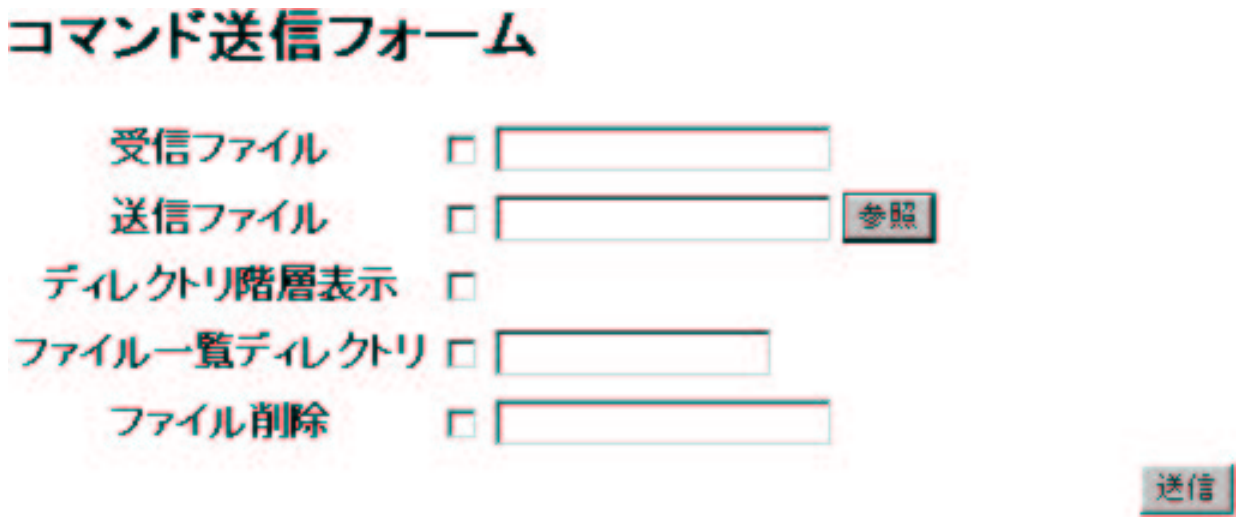

図 4.4　コマンド送信フォーム

コマンド送信フォームには，ユーザに多くの権限を与えず，また不必要なコマンドが実行されないよう，ファイル操作に必要なコマンド 5 個だけしか表示していない．ユーザは，このフォームでファイル操作命令をチェックボックスから選択することで，ファイル操作命令をファイルサーバへ送信することができる．操作命令は，送信時に暗号化されファイルサーバへと送信される．各ファイル操作命令方法を 4.2.5 章より説明する．

4.2.5　受信ファイル

ユーザは，ファイルサーバにあるファイルを受信したい場合，このチェックボックスを選択する．そして，右隣にあるテキストボックスの中に絶対パスで記入し，送信ボタンを押すことでファイルサーバへ命令を送信できる．送信例を図 4.5 に示す．

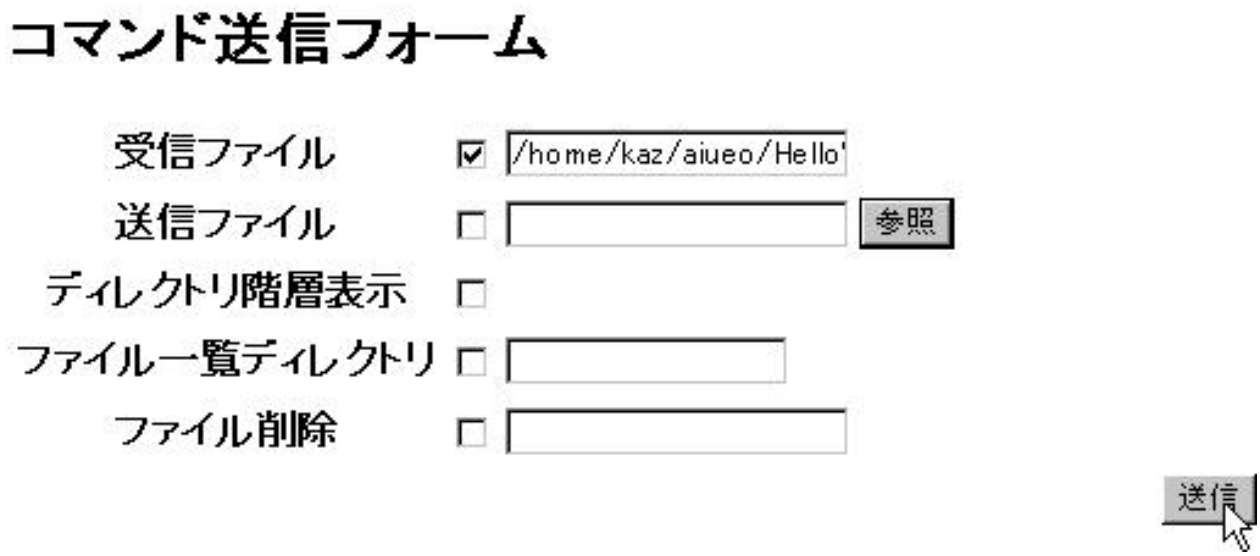

図 4.5　ファイル受信命令送信例

4.2.6　送信ファイル

ユーザは，ファイルサーバにファイルを送信したい場合，このチェックボックスを選択する．そして、右側にある参照ボタンを押すことで図 4.6 のようなファイルダイアログボックスが開き，ファイルを選択することで図 4.7 のように右隣にあるテキストボックスにファイル名が入るようになっている．

図 4.6　ファイルダイアログボックス画面

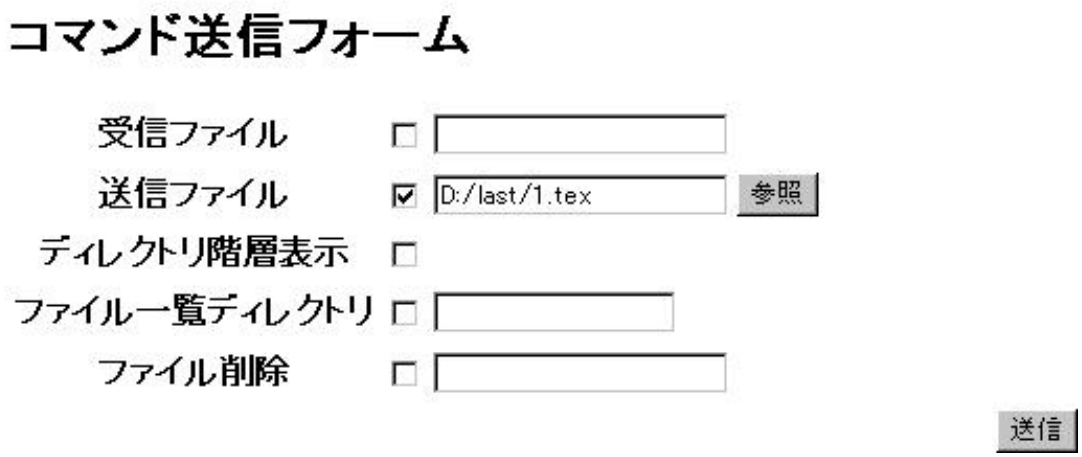

図 4.7　送信ファイルがテキストボックスに入った例

4.2.7　ディレクトリ階層表示

ユーザは，ファイルサーバの自分のホームディレクトリ以下のディレクトリ階層を調べたい場合，このチェックボックスを選択する．

4.2.8　ファイル一覧ディレクトリ

ユーザは，ファイルサーバ内のファイルの一覧を参照したい場合，このチェックボックスを選択する．そして，参照したいディレクトリを絶対パスで記入し，ファイルサーバへ命令を送信する．

4.2.9　ファイル削除

ユーザは，ファイルサーバにあるファイルを削除したい場合，このチェックボックスを選択する．そして，右隣にあるテキストボックスの中にファイル名を絶対パスで記入し，ファイルサーバへ命令を送信する．

4.3　結果の表示

結果の表示の部分は，ブラウザの下半分を利用し図 4.8 のように表示する．結果の表示部分は２つにわけ，左側に受信メッセージ (サーバから送られた結果の HP) の一覧とともにタイムを表示し，メッセージ選択したり，削除できるようにしている．そして，残りの部分にメッセージ内容を表示できるようにしている．

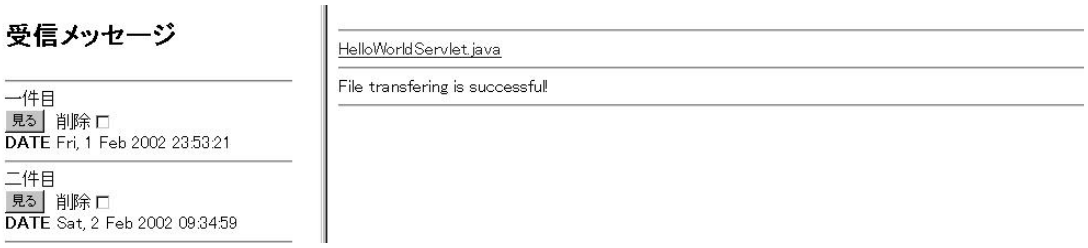

図 4.8　メッセージの表示画面

4.3.1　メッセージの削除

ユーザは，メッセージを削除したい場合，図 4.9 のようにメッセージ一覧の中から削除したいメッセージの削除のチェックボックスを選択し，コマンド送信フォームの送信ボタンを押して，ファイルサーバに命令を送信すると，図 4.10 のようにメッセージが削除される．

図 4.9　メッセージの削除前

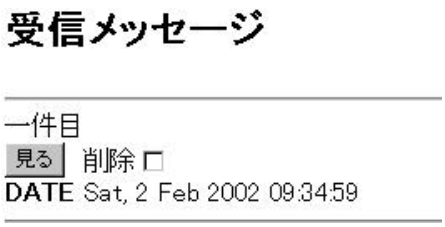

図 4.10　メッセージの削除後

4.4　ファイルサーバの操作処理

出先端末から送られてきたファイル操作命令は，ファイルサーバで命令を復号し JavaServlet からサーバのシェルプログラムに引数を渡し，実行することで，ファイル操作処理を行う．ファイル操作処理は，ユーザの命令ミスによって生じる再送信命令回数を減らすため，各操作命令によってエラー結果を表示するのではなく，再び必要とされるであろう操作命令の結果を表示するようにした．各命令のファイル操作処理は，次より説明する．

4.4.1　受信ファイルの処理

”受信ファイル”のファイルサーバでの処理を図 4.11 に表す.

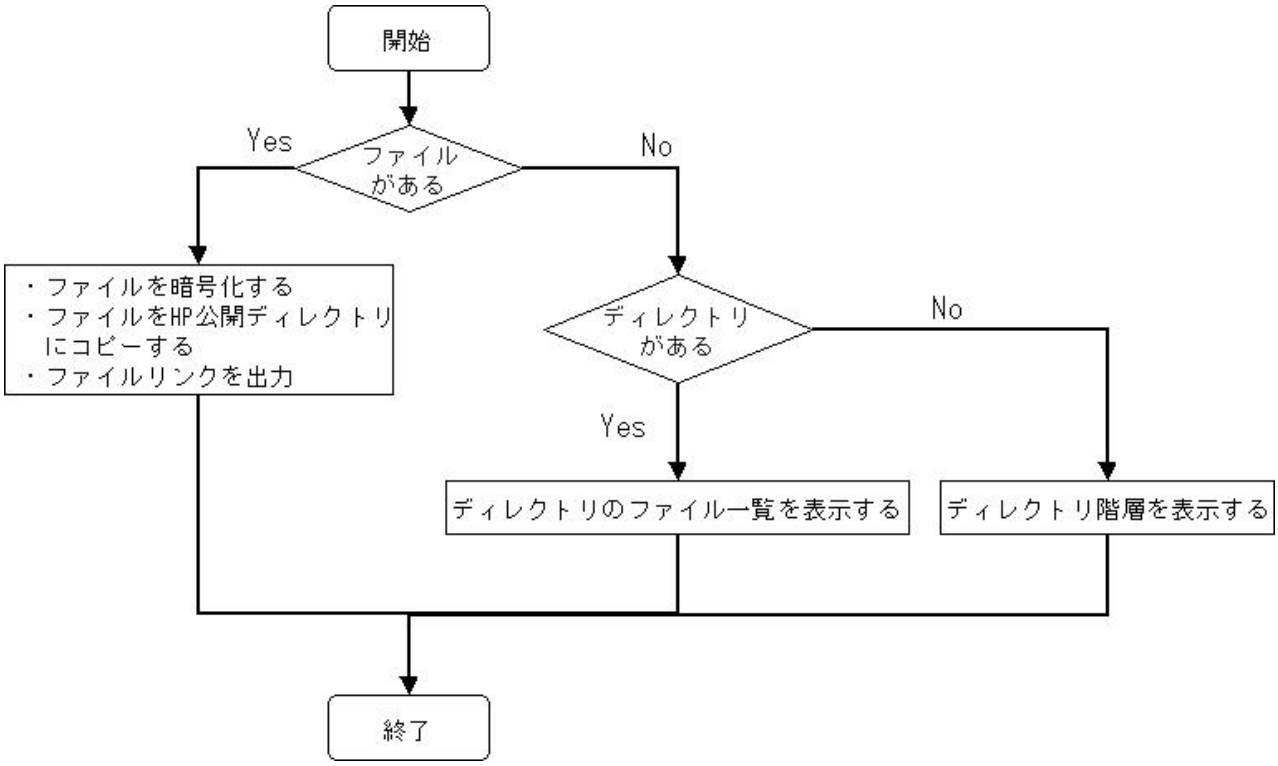

図 4.11　”受信ファイル”のサーバ処理

ユーザから送信された命令より，指定されたファイルがあるかを判断する．次に，指定されたファイルがある場合，そのファイルを今回の SAS 認証に用いた認証情報を用いて暗号化する．そして，暗号化したファイルを HP の公開ディレクトリにコピーする．最後に，ファイルへのリンクを張った HP を作成し，ユーザに結果として返す．また，ファイルが存在しない場合には，指定されたファイルのディレクトリがあるかを判断する．ディレクトリがある場合は，”ファイル一覧ディレクトリ”の結果をユーザに返す．ディレクトリがない場合は，”ディレクトリ階層表示”の結果をユーザに返す.

”受信ファイル”の操作命令が成功した場合の HP の結果例を図 4.12 に表す.

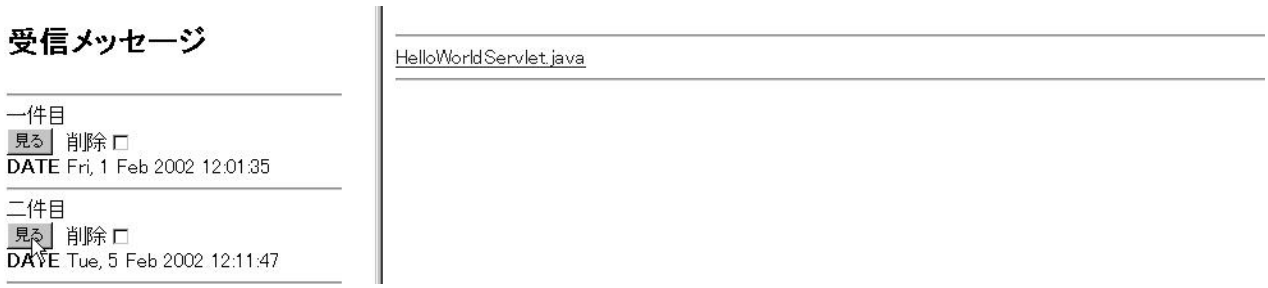

図 4.12　”受信ファイル”操作成功例

その後，ユーザはブラウザより DownLoad のコマンド (図 4.13) をすることで出先端末の指定した場所にファイルを保存する (図 4.14) ことが可能である．

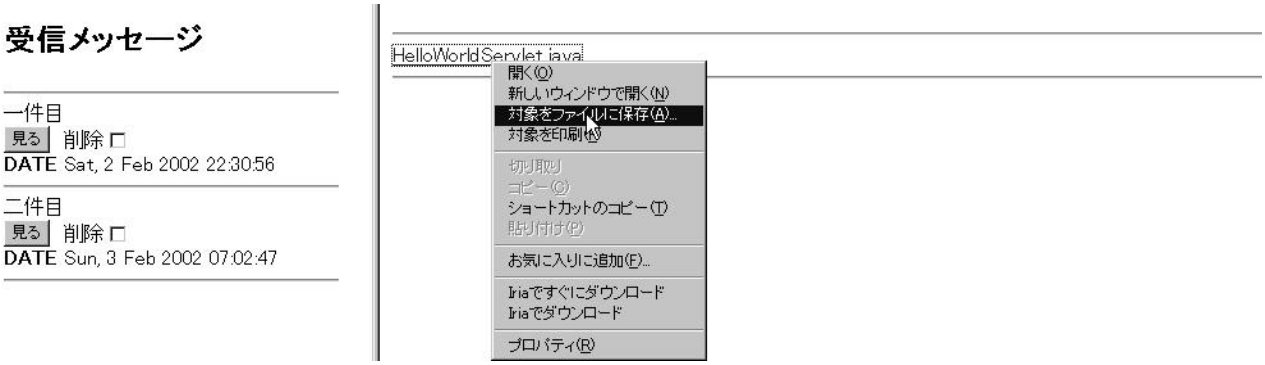

図 4.13　DownLoad の実行

図 4.14　ファイルの保存

4.4.2　送信ファイルの処理

”送信ファイル”のファイルサーバでの処理を図 4.15 に表す.

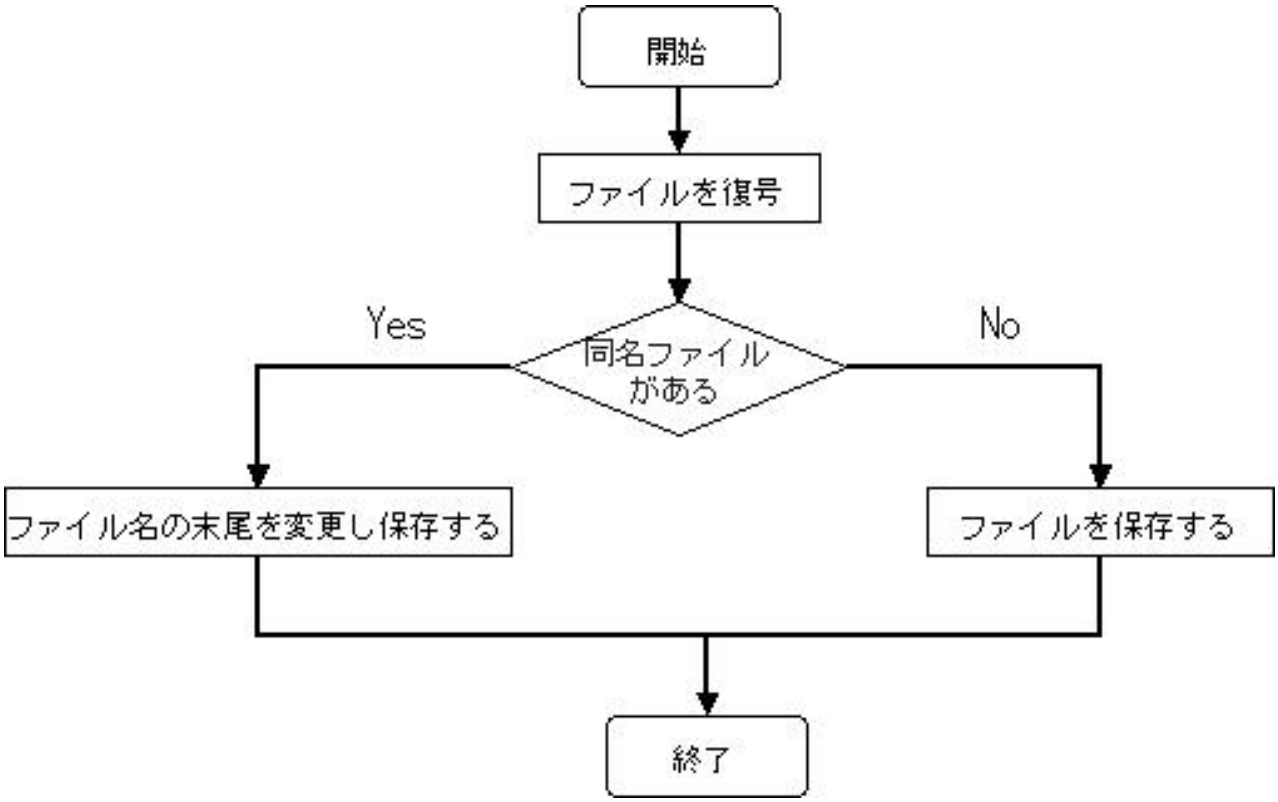

図 4.15　”送信ファイル”のサーバ処理

ユーザから送信された命令とファイルを複合する．複合されたファイルは，ユーザのホームディレクトリ以下の特定のフォルダに保存される．しかし，同名のファイルが存在した場合には，ファイルの上書きを行わず，ファイル名の末尾を変更し保存させるようにする．

”送信ファイル”が成功した場合の HP の結果例を図 4.16 に，失敗した結果例を図 4.17 に示す.

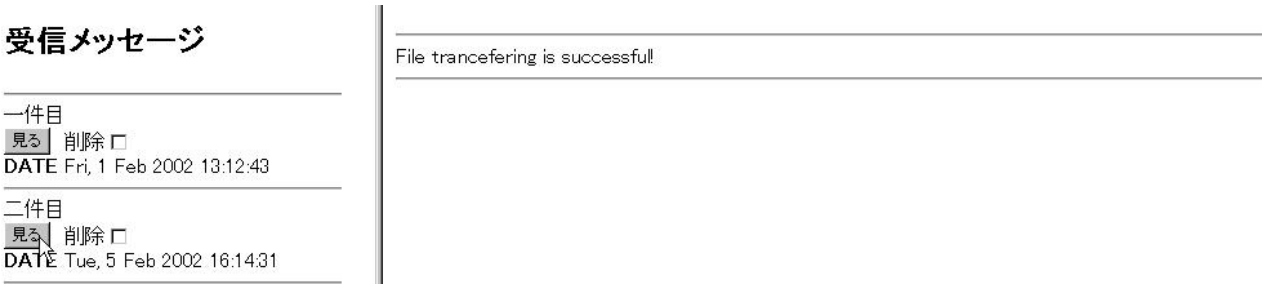

図 4.16　”送信ファイル”操作成功例

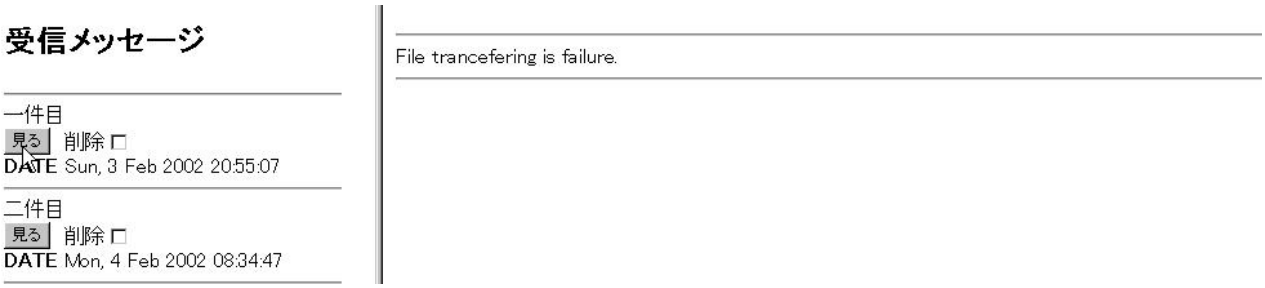

図 4.17　”送信ファイル”操作失敗例

4.4.3　ディレクトリ階層表示の処理

”ディレクトリ階層表示”のファイルサーバでの処理は，ユーザのホームディレクトリ以下のディレクトリ階層を HP に出力する．HP に出力される結果例を，図 4.18 に表す.

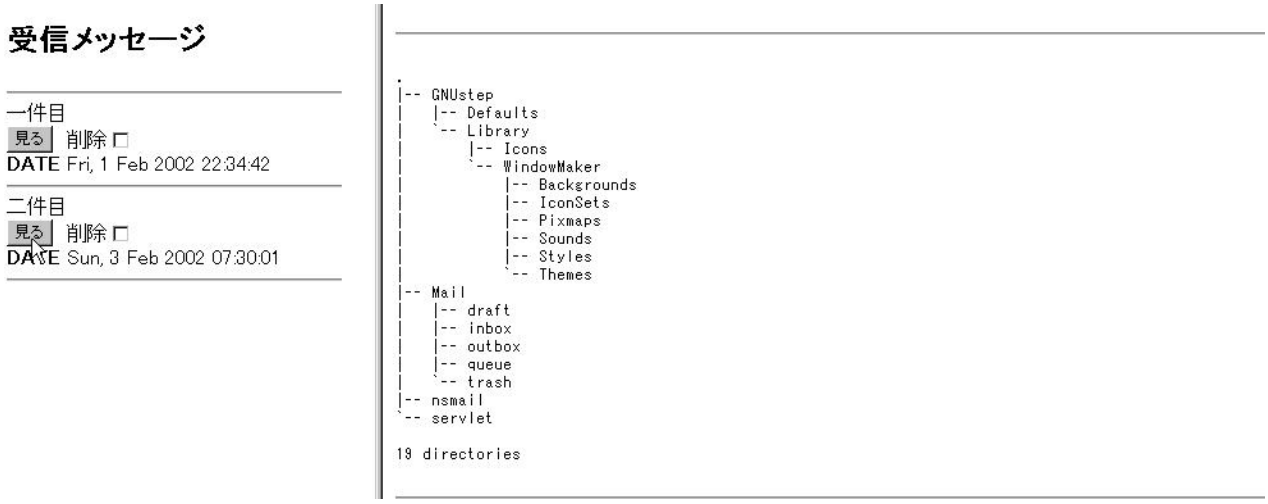

図 4.18　”ディレクトリ階層表示”操作成功例

4.4.4　ファイル一覧ディレクトリの処理

”ファイル一覧ディレクトリ”のファイルサーバでの処理を図 4.19 に表す.

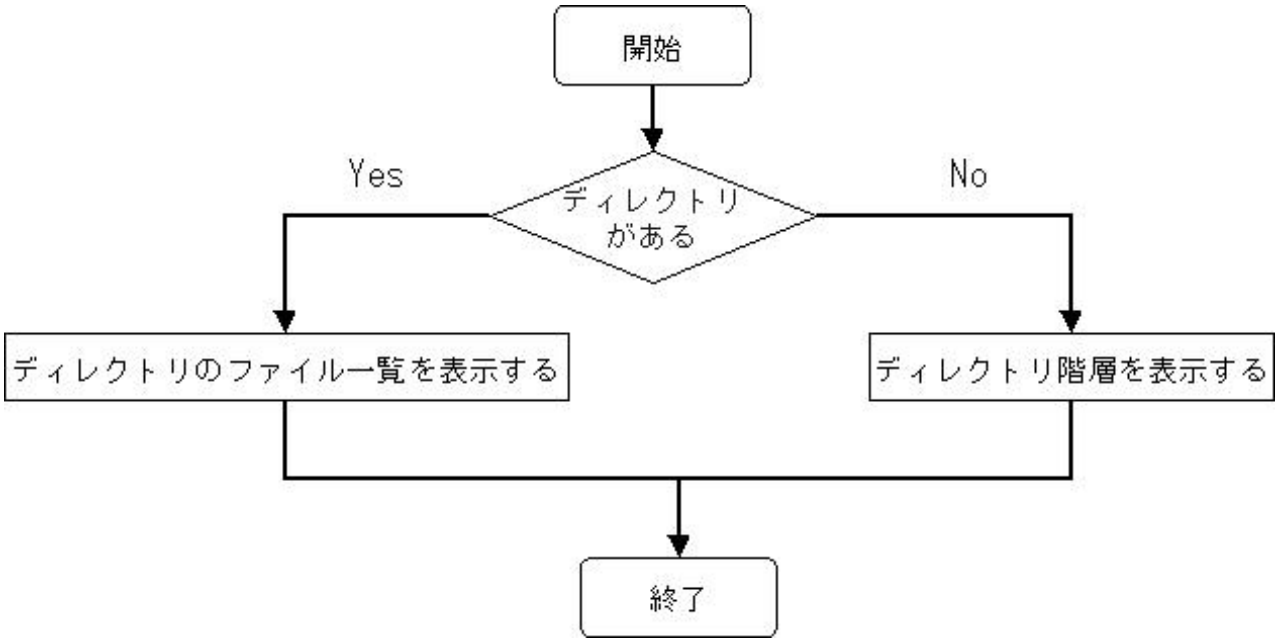

図 4.19　”ファイル一覧ディレクトリ”のサーバ処理

ユーザから送信された命令より，ファイルの一覧を表示するディレクトリがあるかを判断する．ディレクトリがあれば，ディレクトリのファイル一覧の結果の HP を作成し．ディレクトリがない場合には，4.4.3 章で述べた”ディレクトリ階層表示”の結果を返す．”ファイル一覧ディレクトリ”の操作命令が成功した場合の HP の結果例を図 4.20 に表す.

図 4.20　”ファイル一覧ディレクトリ”操作成功例

4.4.5　ファイル削除の処理

”ファイル削除”のファイルサーバでの処理を図 4.21 に表す.

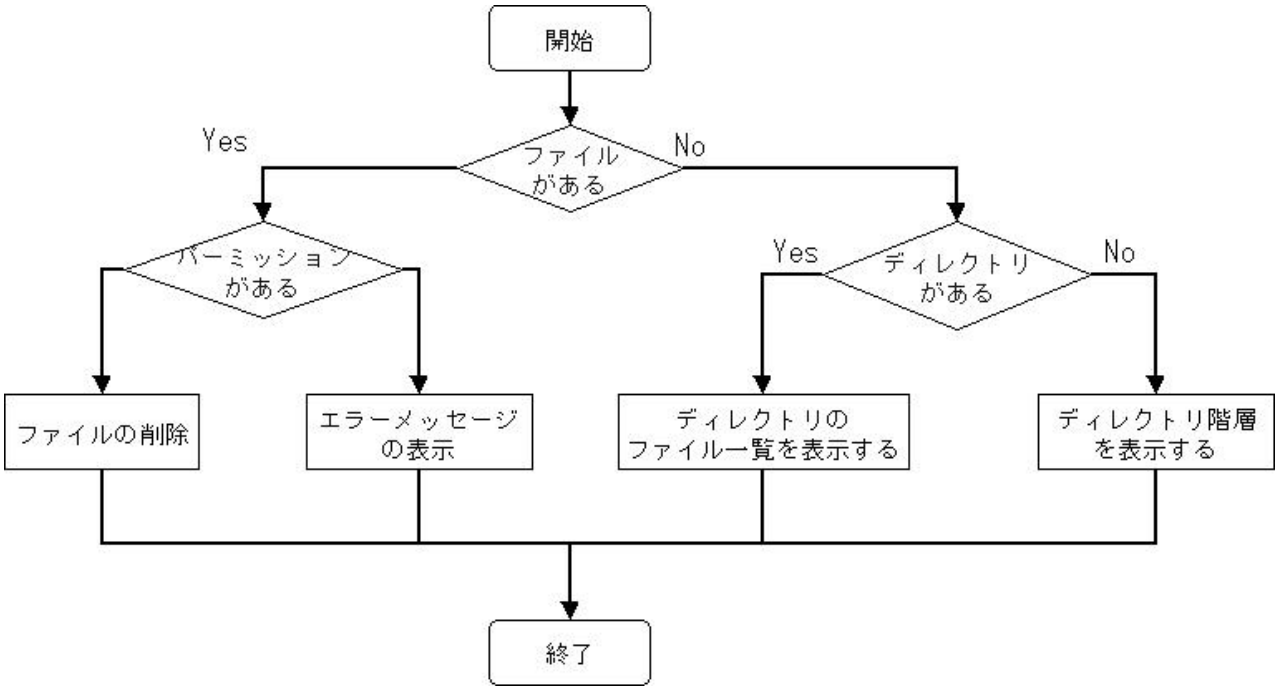

図 4.21　”ファイル削除”のサーバ処理

ユーザから送信された命令より，指定されたファイルがあるかを判断する．次に，指定されたファイルがある場合，ユーザがファイルのパーミッションを持っているかどうか判断し，持っていた場合はファイルを削除して図 4.22 のようなメッセージ結果を返す．ファイルのパーミッションがない場合，図 4.23 のようなエラーメッセージを出力する．また，ファイルが存在しない場合には，指定されたファイルのディレクトリがあるかを判断する．ディレクトリがある場合は，”ファイル一覧ディレクトリ”の結果をユーザに返す．ディレクトリがない場合は，”ディレクトリ階層表示”の結果をユーザに返す.

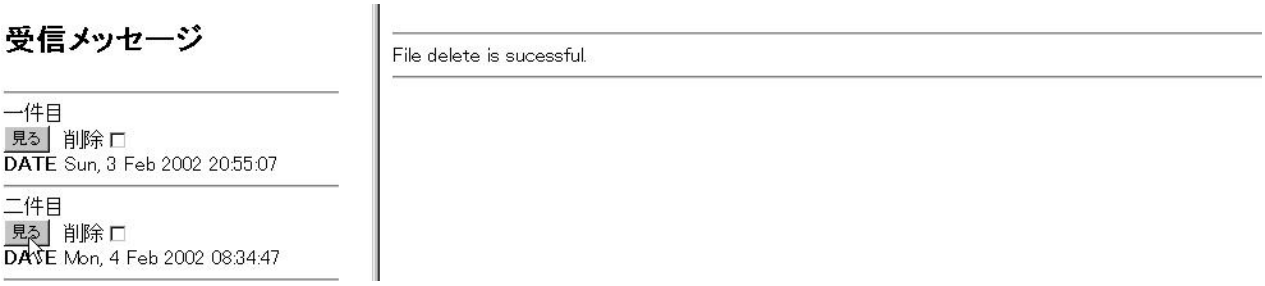

図 4.22　”ファイル削除”成功例

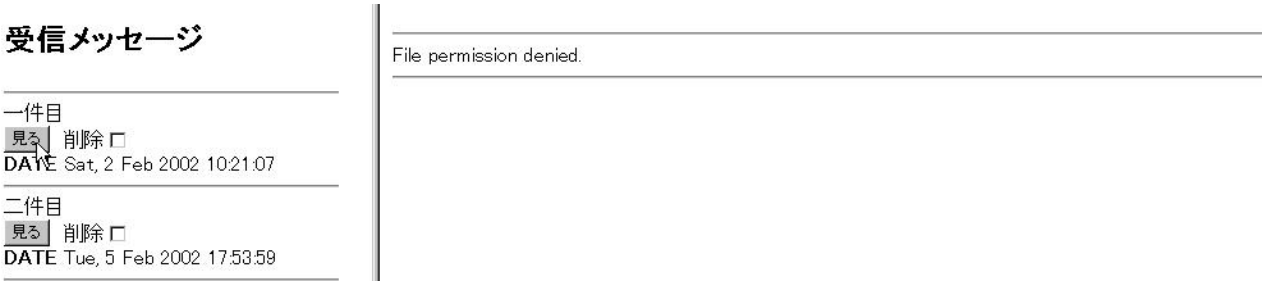

図 4.23　”ファイル削除”エラーメッセージ

第 5 章

今後の課題

本研究では，試作システムを構築したが，完全な構築にはいたっておらず以下のような問題点がある．

- SAS の実装
- 出先端末のローカルファイルへのアクセス
- 復号処理プログラムの配布方法

1 点目は，ログインフォームの認証処理に SAS を実装するだけで良いと考えられる．2 点目については，電子署名付きアプレットを用いることで解決できるようになる．3 点目については，Java Plug-in を利用し Web ブラウザにプログラムを DownLoad させることで解決できる．これらを改善した試作システムを導入し，運用面での評価が必要である．評価項目については，次のようなものが考えられる．

- 通信トラブルからの回復手段
- サーバへの負荷実験
- FTP とのファイル転送効率の比較
- ブラウザインタフェースの評価
- 安全性の評価

1 点目は, 停電や断線といった通信トラブルからの回復手段の確立である．2 点目は，複数のユーザによる同時アクセスの場合にかかるサーバの負荷を測定する必要がある．3 点目は，試作システムと FTP によるファイル送受信における時間を比較する必要がある．4 点目は，

ユーザにとって使いやすいシステムであるかの検証が必要である．5 点目は，実際に運用した点での安全性の評価が必要になる．

第 6 章

まとめ

本研究では，イントラネット外の端末から，イントラネット内のファイルサーバへアクセスするために，HTTP をベースにした出先端末の環境に依存しないセキュアなファイル送受信サービスである，遠隔コンテンツ操作方式を提案し，試作システムを構築した．今後は，試作システムの実用に向けてさらなる改善が必要になる．

謝辞

本研究を行うに際し，多大なるご指導，ご鞭撻を頂いた高知工科大学情報システムの清水明宏教授に深く感謝いたします.

また，本研究室の院生　岡田　実氏，田鍋　潤一郎氏，辻　貴介氏，ならび学部生，伊藤哲君，植田　洋加さん，越智　裕架子さん，上岡　隆君，河村　智君，木村　大樹君，窪内美紀さん，中務　秀和君，その他大学生活の中で私を支えてくださった皆々様深く感謝をいたします.

参考文献

[1] M.Sandirigama, A.Shimizu, M.noda, “ Authenticated and Secure Communication over Insecure Channels - Simple and Secure (SAS) Authenticated Password ”, IEICE Trans. Commun., Vol.E83-B, No.6,JUn.2000.

[2] 上岡　隆, 清水　明宏, ”ワンタイムパスワード認証方式 SAS の安全性に関する検討”, 電子情報通信学会技術研究報告書, OFS2001-48, NO. 435, pp. 53-55, 2001.

[3] 堀岡, 戸田, 清水, “ 電子メール転送サービス方式の検討 ”, 信学技法, TECHNICAL PEPORT OF IEICE, OFS97-39, IE97-77(1997-09)

[4] 岡崎　友輝年, “ コンテンツ転送サービス方式 ”, 高知工科大学　情報システム工学科

[5] 谷口　功, “ 図解ネットワークセキュリティ -攻撃と防御のメカニズム- ”, 株式会社オーム社

[6] 原田　洋子, “ Java Servlet 最新サーバ・プログラミング ”, 株式会社秀和システム