平成 15 年度

修　士　論　文

題　目　　ステガノグラフィを用いた
文書流出のリアルタイム検出

学籍番号　　S029321

氏名　　高木　明

平成 14 年度入学

島根大学大学院　総合理工学研究科博士前期課程

数理・情報システム学専攻　計算機科学講座

指導教官　　田中　章司郎　教授

平成 16 年 2 月 20 日受理

目次

第１章　序論

今日の情報化社会の中, 公的機関や大企業はもちろん中小企業でもインターネットに接続するのは当然のようになってきた. しかしそれに伴い内部機密情報や顧客情報の漏洩が深刻な問題となっているのも事実である.

そして既にその問題を解決する手段として, サーバ上のデータベースを暗号化し仮に文書が漏洩しても閲覧不可能にするシステムや, ネットワーク上に流れるデータを常に監視することで漏洩検知するシステムが開発されている. しかし前者のシステムでは文書が暗号化されているため, クライアントで文書を閲覧するためには文書の復号が必要となり, 文書の閲覧に時間がかかる. また後者のシステムでは専用のネットワーク監視機器を使用するか, クライアントマシンに監視アプリケーションをインストールするため, 利用者に監視の存在を認知される. またそれによって心理的圧迫感を与える恐れがある. またその検知方法も禁止文字を全文検索するため, 禁止文字が多いほど検索にかかる負荷が大きくなるなどまだ問題も多い.

そこで本研究では専用の機器を使用せず, 利用者に認知されにくい文書流出の検知システムを開発, 試作することにした. 本研究で使用する機密文書は, 社内統一定型文書規格としてステガノグラフィを用いて署名情報を隠蔽した会社のロゴ画像を文書に挿入する. これによって利用者に気付かれずに署名情報を付加する. そしてパケットキャプチャ機能を持ったネットワーク監視モニタを用いてリアルタイムにネットワークを監視し, パケットレベルに分割された文書ファイルのロゴ画像から署名情報を抽出することで, 文書流出の検出を行うシステムを開発した. 本システムではロゴ画像に署名情報を隠蔽したことにより, 文書内からロゴ画像のみを検索するのみで機密情報か否かを判別でき, 全文検索システムに比べ, 検索にかかる負荷を大幅に軽減可能となる.

本研究ではその試作として機密文書に Microsoft Word 文書を, ロゴ画像に JPEG 画像を採用し, ステガノグラフィを用いた JPEG 画像への署名情報隠蔽及び抽出プログラム, パケットキャプチャ機能を持つモニタプログラムを作成し, FTP および SMTP による機密文書流出の検知実験を行った. またモニタの追加機能として FTP のファイル転送停止機能を作成し, その機能を検証した.

第２章　文書流出検知システム

2.1　システム概要

本文書流出検知システムは内部機密文書を保管するサーバと, その転送を監視するモニタマシンから構成される.

サーバマシンに置かれる文書は社内統一定型文書規格としてステガノグラフィを用いてシ署名情報を隠蔽した会社のロゴ画像が文書の冒頭に挿入されている．署名情報は”top secret”, “classified”, “leak detect”などのその文書の重要度を表す文字列である. 本システムのモニタプログラムは, この署名情報の重要度に応じて文書流出時の対応を変化させることができる.

モニタマシンはシェアードハブ(リピータハブ)を用いてサーバマシンのパケットが監視できる位置に接続される（図 2.1）．また別の接続方法として，１台のマシンに２枚の NIC(Network Interface Card)を挿し，NAT(Network Address Translation)や IP マスカレードを構築してもサーバのデータは監視可能である（図 2.2）.

モニタプログラムはパケットキャプチャ機能を用いてリアルタイムにパケットを取得する. 取得されたパケットは解析され必要な情報を取得し, 分割されたパケットから直接署名情報を抽出する. そしてロゴ画像に隠蔽された署名情報から文書の流出を検知する. 署名情報から抽出される文書の重要度に応じた対応を行う. 本研究では試作として FTP 及び SMTP を実装している. また流出後の対応として FTP にはファイル転送停止機能を，また FTP，SMTP ともにメール送信機能を追加した.

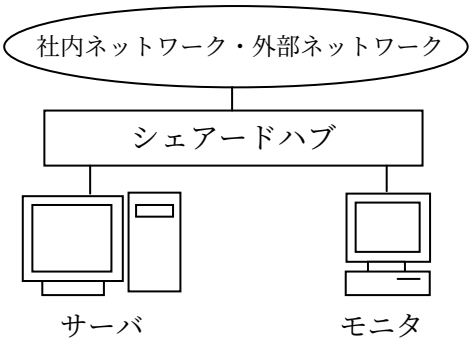


図 2.1　シェアードハブを用いた接続

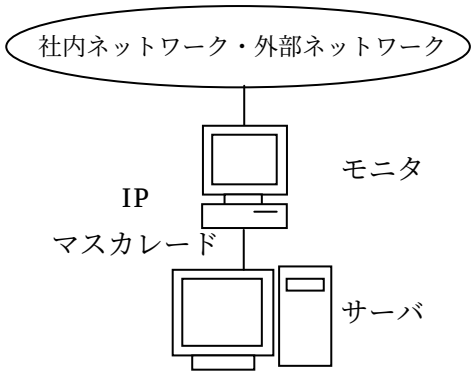


図 2.2　IP マスカレードを用いた接続

2.2　先行研究との比較

先行研究の調査として以下の３つのサイトで検索を行った.

- 文献データベース INSPECC

Steganography & packet	0 件
Steganography & leak & detect	0 件
Steganography & classified & document	1 件 *1

*1 ステガノグラフィの解析の困難性に関する研究

- 特許庁　特許電子図書館

ステガノグラフィ & 漏洩	1 件
ステガノグラフィ & 検出	9 件
ステガノグラフィ & パケット	1 件
ステガノグラフィ & 機密文書	0 件

- Google

ステガノグラフィ & 漏洩	38 件
ステガノグラフィ & 検出	234 件
ステガノグラフィ & パケット	94 件
ステガノグラフィ & 機密文書	1 件 *2

*2 ニュースサイトの関連リンク集にヒットした

これらの検索結果で文書流出のリアルタイム検出にステガノグラフィを用いたものは無い．しかし実際に開発されたシステムは存在するため，本システムと同様の機能を持つ実製品として全文検索システムとの比較を行う．

禁止文字を全文検索するシステムの場合，禁止文字の個数に比例して検索に負荷がかかるという難点がある．また禁止文字の選択方法によっては，誤検知が発生する可能性がある．それに対し，本システムはキーワードを全文検索するが，そのキーワードはJPEGの開始マーカひとつであるため，上記の全文検索に比べて検索にかかる負荷が圧倒的に少ないという利点がある．またキーワードを誤検知してもモニタプログラムはJPEGの構造を理解するため，そのまま解析されることはない．

第３章　社内統一定型文書規格

本システムで扱う文書は社内統一定型文書規格として文書の先頭に会社のロゴ画像が挿入されているものとする．使用する文書ファイルは，その普及度から Microsoft Word 文書を使用することとした．しかし Microsoft Word 文書は文字，画像，その他のオブジェクトは圧縮されて保存されているため, 閲覧するにはそれを復元する必要がある．しかし本研究を通じて，その圧縮方法を知ることはできなかった．そこで実験的に Word 文書の構造を調べることにした.

3.1　画像の選択

まず，画像の圧縮方法を知るため，文書ファイルにさまざまな形式の画像を挿入し，それらが圧縮されるかを調べた.

実験１

バイナリエディタを用いて，画像と同様のビット列が Word 文書内で現れるか確認.

実験の結果, JPEG のみ Word 文書ファイル内でも圧縮されずにそのまま保存されることが判明した.（表 3.1）これによって Word 文書ファイル内から直接 JPEG の開始マーカが検索できることになり，JPEG 画像の検出が容易となったため，本研究で用いるロゴ画像を JPEG 画像とした.

表 3.1　各圧縮形式の可逆／不可逆圧縮と Word での圧縮

圧縮形式	可逆／不可逆圧縮	Word 圧縮
BMP	未圧縮	圧縮
JPEG	不可逆圧縮	未圧縮
GIF	可逆圧縮	圧縮
PNG	可逆圧縮	圧縮
TIFF	未圧縮／可逆圧縮／非可逆圧縮	圧縮

3.2　Word 文書の構造

Word の先頭にはヘッダが入っていることは良く知られていたが，後に続くデータの構造が不明のため次のような実験を行い，JPEG 画像が Word ファイル内のどこに挿入されるかを調べた.

実験２

文書ファイルに含まれる文字数を変化させ，そのとき JPEG 画像の位置がどのように変化するかをバイナリエディタで確認する．

この実験の結果，文字数が増えるにつれ JPEG 画像がファイル内の後方に保存されることから，文字部分は画像部分の前に保存されることがわかった．また文字数が変化してもあまりビット列の内容と位置が変化しない部分，つまりヘッダとフッタの存在が確認できた．

最後に画像の保存順序について実験を行った．

実験３

１）画像の挿入順を変更する

２）画像の保存順を変更する

３）画像のサイズを変更する

実験２と同様，バイナリエディタを用いて，その変化を確認する．

その結果，ファイル中で画像が出現する順番には挿入順，保存順，サイズは関係がなく，行に関して先頭にあるものから順に保存されることが分かったその挿入位置のみによって決まることがわかった．

次に，画像の書式指定によって変化するかについて実験を行った．

実験３

４）レイアウト指定を変更する

実験３，４の結果より以下のことが分かった．（図 3.1）

①行内指定の画像はほかの指定の画像の後に保存される

②四角，外周，背面，前面には強弱はなく，デフォルトの行内の指定からそれらの配置設定に変更された順に保存される．その後文書中の任意の場所に移動させてもその出現順は変わらない

つまり Word 文書ファイルの最初に出現する画像がロゴ画像になるためには，社内統一定型文書規格のテンプレートの段階でロゴ画像が四角，外周，背面，前面のいずれかで挿入されていなければならない．ただしロゴ画像の書式指定が１度でも行内に変更されると，他に四角，外周，背面，前面の指定がされている画像があった場合，それらが最初に保存されるようになるため，ロゴ画像が最初に出現しなくなる．

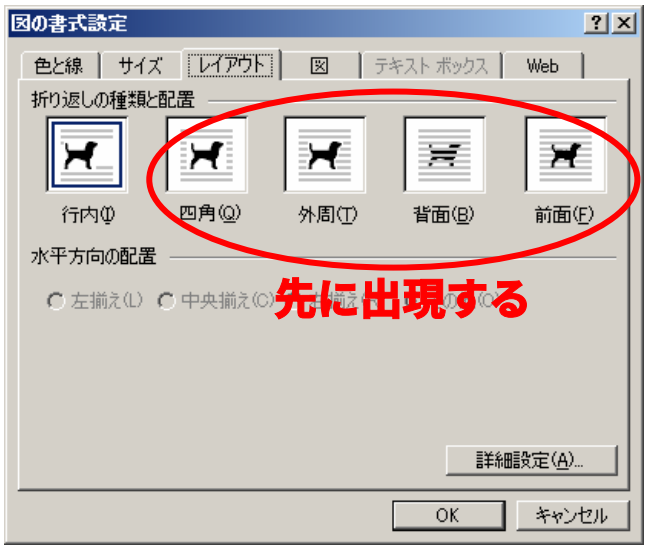


図 3.1　図の書式設定

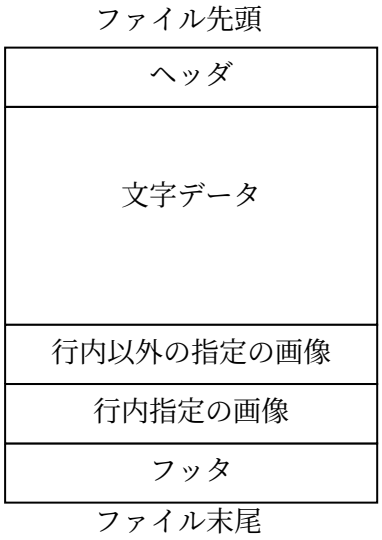


図 3.2　Word 文書の構造

実験の結果より「テンプレートに手を加えない」という条件の下では，ロゴ画像はファイルの最初の画像として出現させることはできることが確認できた．

本システムではそれを利用して，最初に現れる画像，つまりロゴ画像を解析した後，以降の解析を打ち切る．これによって処理時間を軽減する．

しかし社員によってロゴ画像のレイアウト指定が解除された場合，ロゴ画像が最初に出現しない．そこまで考慮する場合は打ち切りを行わないほうが良いと考えられる．

3.3　Word 文書からの JPEG 画像の抽出方法

先の実験によって JPEG は Word ファイル内でも圧縮されないことが判明した．そこで本システムは，Word ファイル内から JPEG の開始マーカを全文検索する．

JPEG の開始マーカ

マーカ	シンボル	意味
FFD8	SOI	イメージ開始(Start Of Image)

本来は FFD8 を検索するべきであるが，

- SOI は他の情報を持たないため，マーカの後には次のマーカが続く
- マーカは必ず FF で始まる（付録 1）

ことを利用し，本システムでは FFD8FF を全文検索する．これによって FFD8 を全文検索するよりも誤検知の確率を減らすことができる．また誤検知しても本システムは JPEG の構造を解析するため，そのまま解析が続くことは無い．

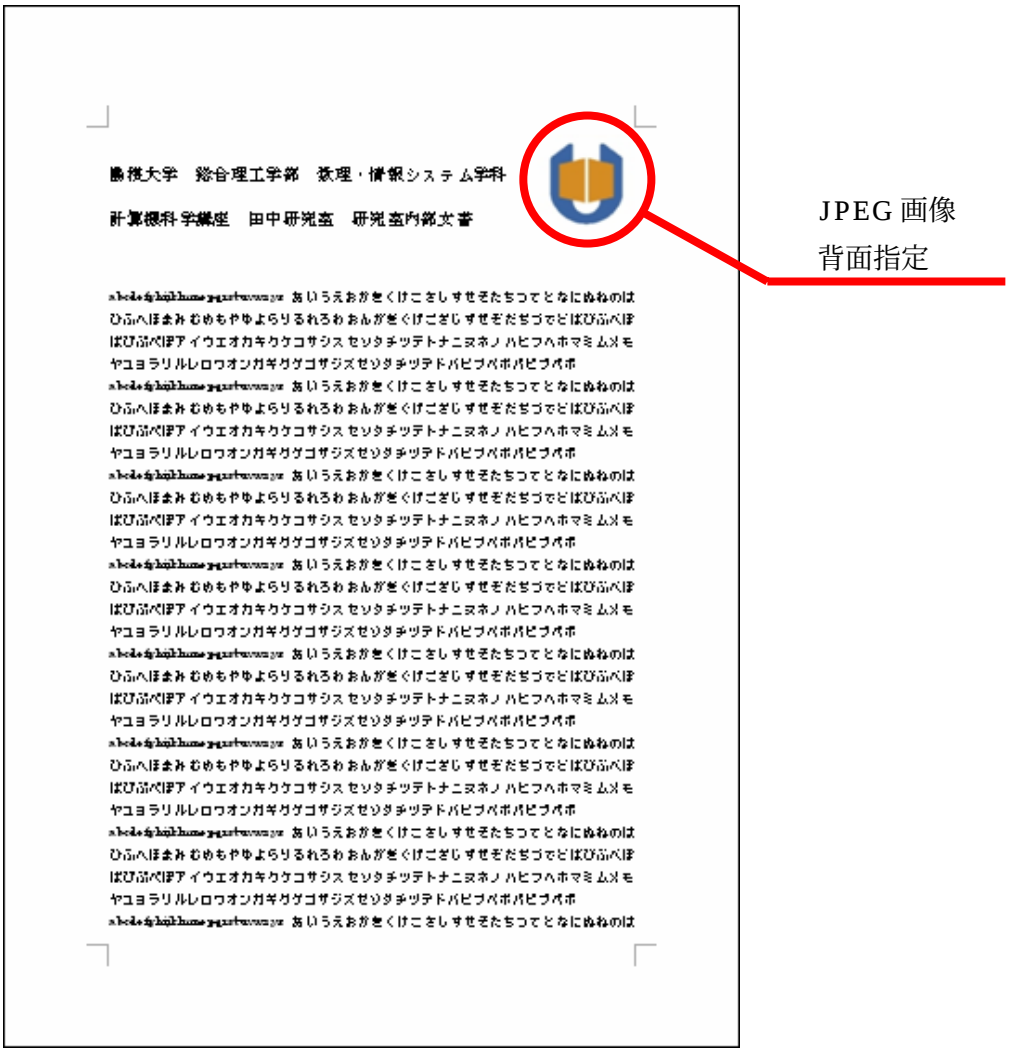


図 3.3　社内統一定型文書規格

3.4　文書の重要度

本システムでは文書の重要度を３種類用意した. 文書にどの重要度を与えるかについては，本研究では議論していない.

- レベル１

意味　　比較的重要度の低い文書　広報など
隠蔽文字　　leak detect
流出時の対応　管理者にメールで通知

● レベル２

意味　　中程度の重要度の文書　社内新聞など
隠蔽文字　　classified
流出時の対応　管理者，役員にメールで通知

● レベル３

意味　　最も重要度の高い文書　企画書，顧客情報など
隠蔽文字　　top secret
流出時の対応　FTP の場合，直ちに転送停止
　　管理者，役員にメールで通知

なお管理者, 役員のメールアドレスはあらかじめコンフィグファイルにて設定しておく.

第４章　ステガノグラフィの紹介

本システムの署名情報の隠蔽に用いているステガノグラフィについて紹介する．ステガノグラフィの詳細については参考文献[1][2][3]を参照されたい．

4.1　ステガノグラフィとは

通常の暗号化は秘密にしたい情報を暗号化アルゴリズムを用いて暗号文に変換する．しかし暗号化された意味不明な文字の羅列やファイルは，それを見た第三者に，それが暗号化された何らかの情報であることを認知されてしまう．つまり悪意ある第三者から見れば，それは解析の対象となることになる．

それに対し，ステガノグラフィは秘密情報を別のデータ中に埋め込み，外見上はそれが暗号文であることを隠蔽する．そのため一見して暗号であると認知できず，隠蔽の存在そのものを隠蔽することが可能となる．本システムはこの事実を利用して，利用者に意識させずに署名情報を埋め込む．これは利用者に監視の存在による心理的圧迫感を与えないためである．

4.2　隠蔽アルゴリズム

本研究では周波数領域における量子化誤差を利用する方法のひとつとして，量子化テーブルへの隠蔽を採用した．この方法では隠蔽可能なビット数は少ないが画像に手が加えられない限り，必ず復号できるという利点がある．本研究では署名情報としてキーワードを隠蔽するため，大量の隠蔽領域は不要であるからこのデメリットは問題ない．またこの方法の特徴として抽出が簡単という点がある．通常のステガノグラフィの考え方では，頑健性の低い隠蔽方法ということになるが，すべての処理をリアルタイムで行う本システムにおいては処理時間がかからないという点で有用である．

今回採用したアルゴリズムは参考文献には記載されていなかったため，先行研究について以下のキーワードを用いて検索した．

INSPECC

steganography ＆ quantization table	0 件

特許庁電子図書館

ステガノグラフィ ＆ 量子化テーブル	0 件

Google

ステガノグラフィ ＆ 量子化テーブル	0 件
steganography ＆ quantization ＆ table	542 件

Google では，量子化テーブルを利用したより高度なステガノグラフィが公開されていた．よってそれより単純なこの隠蔽方法は既知の技術であると思われる．

隠蔽アルゴリズムの概観は図 4.1 のようになる．

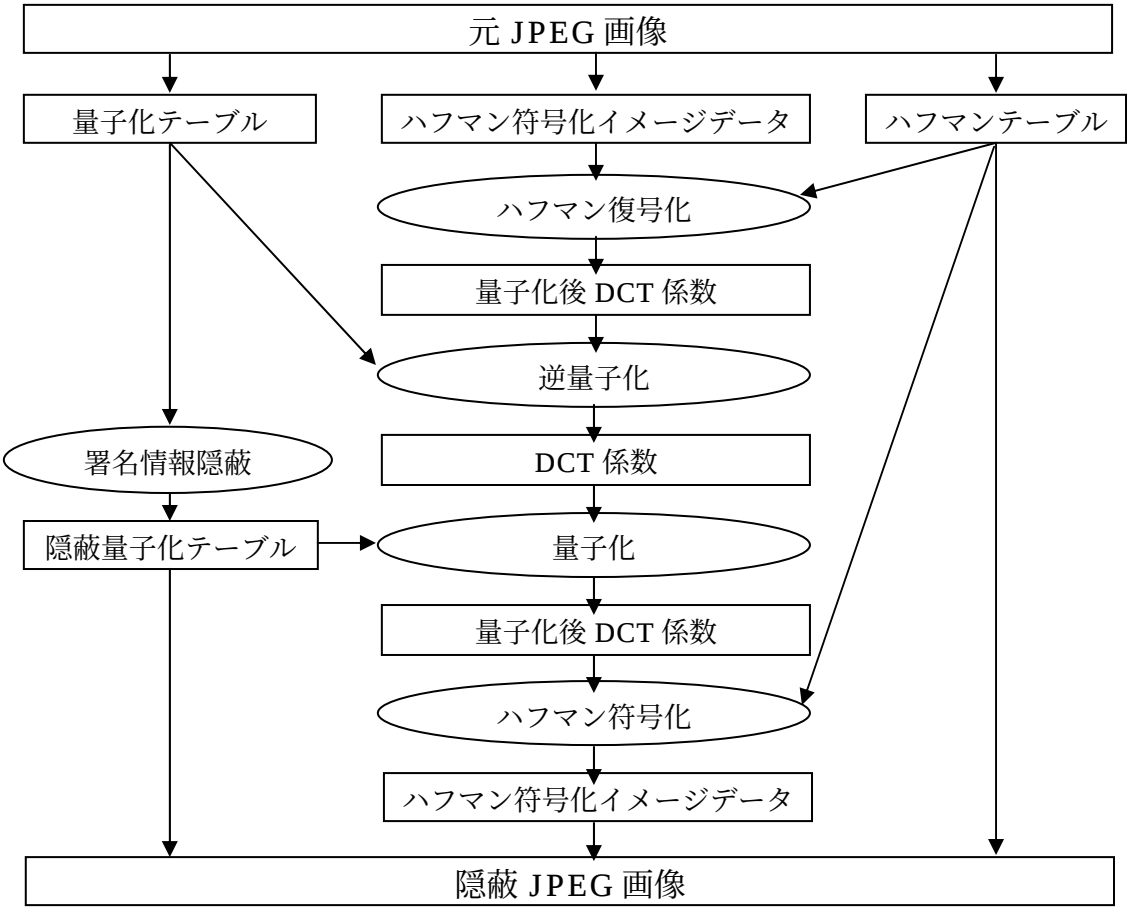


図 4.1　署名情報隠蔽の流れ

会社のロゴや証印など文書にすべての文書に付加しても違和感のない画像を用意する．この隠蔽方法はスタンダード方式の 24bit フルカラーJPEG 形式にしか適用できないため，形式が異なる場合は 24bit フルカラーJPEG 形式に変換する必要がある．また文書ごとに画像を変更してもかまわない．

隠蔽には JPEG 圧縮されたイメージデータを量子化前の DCT 係数にまで復号する必要がある．JPEG 圧縮の詳細については参考文献[4][5]および ISO/IEC 10918-1～4, (ITU-T Recommendation T.81～84)を参照されたい．

まず JPEG 画像の各セグメントより量子化テーブル，ハフマンテーブル，及びその他の復号に必要な情報を取り出す．そのあと JPEG の復号ステップのハフマン復号化，逆量子化を行う．これによってイメージデータは DCT 係数にまで変換される．

その後量子化テーブルに対して署名情報の隠蔽を行う．隠蔽は量子化テーブルの各要

素の最下位１ビットに対し, 画素置換型ステガノグラフィと同様の方法を用いて隠蔽を行う.

隠蔽する文字列が”top secret”の場合，最初の “t”は ASCII コードで 0111 0100 であるから，これを量子化テーブルの最下位ビットと置き換える（図 4.2）.

隠蔽前量子化テーブル（抜粋）

16	11	12	14	12	10	16	14	
0001 0000	0000 1011	0000 1100	0000 1110	0000 1100	0000 1010	0001 0000	0000 1110	
0	1	1	1	0	1	0	0	隠蔽文字 ”ｔ”
0001 0000	0000 1011	0000 1101	0000 1111	0000 1100	0000 1011	0001 0000	0000 1110	
16	11	13	15	12	11	16	14	

隠蔽後量子化テーブル

図 4.2　隠蔽の様子

以上のように１文字隠蔽するために８個の量子化テーブルの要素が必要となる. 量子化テーブルは８ｘ８であるので１枚の量子化テーブルに８文字が隠蔽可能となる. 量子化テーブルは通常のカラー画像では輝度と色差の２枚が存在するので, 合計１６文字の隠蔽が可能となっている．また文字数が 16 に満たない場合は NULL（0000 0000）を隠蔽する.

量子化テーブルへの隠蔽後，署名情報が隠蔽されたテーブルを用いて JPEG 画像を再符号化する．つまり量子化，ハフマン符号化を行う．本来ここでハフマンテーブルを再設定する必要があるが,

・一般的にハフマンテーブルはデフォルトのものが使われることが多い

・データの特性がほとんど変わらないためハフマンテーブルに大きな変更は無い

の２点から本システムではハフマンテーブルは元の JPEG 画像のものをそのまま使用して符号化を行う.

ただ量子化テーブルに隠蔽するのでは無く，イメージデータを DCT 係数にまで復号する理由は, 量子化テーブルのみを変更すると実際に画像を表示した際, 元の画像と異なったイメージで再生されるためである.

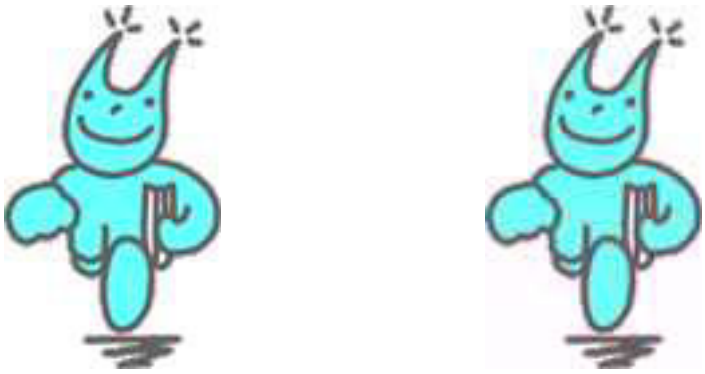
実際署名情報隠蔽を行った画像を示す.

例１

画像作成ソフト　　Paint Shop Pro 7.00

画像サイズ　　81x136

圧縮率(Quality)	50（Paint Shop では圧縮率 50 と表示）
隠蔽文字列	top secret
画像	島根大学マスコットキャラクター「ビビット」



隠蔽前　　隠蔽後

図 4.3　元画像と隠蔽画像の比較 1

例 2

画像作成ソフト	Paint Shop Pro 7.00
画像フォーマット	24bit フルカラーJPEG（スタンダード方式）
画像サイズ	79x79
圧縮率(Quality)	99（Paint Shop では圧縮率 1 と表示）
隠蔽文字列	top secret
画像	島根大学学章

隠蔽前　　隠蔽後

図 4.4　元画像と隠蔽画像の比較 2

4.3　抽出アルゴリズム

続いて抽出アルゴリズムについて説明する．先に説明した隠蔽アルゴリズムは，本システムのモニタプログラムとは別の隠蔽専用のプログラムに搭載されるが，この抽出アルゴリズムはモニタプログラムの署名情報解析部に搭載される．モニタはリアルタイム

で解析を行うため無駄な処理をできる限り省くように設計した.

本研究で採用したステガノグラフィはイメージ部分に手は加えるものの, 隠蔽された文字自体は量子化テーブルに保存されている. そのため抽出の際にはイメージデータなどを必要とせず, 量子化テーブルのみを解析することで署名情報を抽出することが可能となっている．この点は先にも述べたとおり，処理時間の軽減を可能にしている.

JPEG のヘッダは<SOI>,<EOI>はマーカのみから構成され，それ以外のマーカは情報の長さとマーカ情報から構成されている（付録１）．そこで DQT（量子化テーブル定義セグメント）以外のセグメントは，マーカの次に登場するそのセグメントの長さを取得し，その長さだけ読み飛ばす．このセグメントの長さはマーカを除く情報の部分の長さであり，長さ情報の部分も含むため取得された長さから-2 バイト数だけ読み飛ばす.

DQT セグメントが出現するとそれを保存する．量子化テーブルが２枚出現すると即座に解析が行われ量子化テーブルから署名情報の抽出を行う.

抽出では隠蔽とは逆に最下位ビットを順に取り出しそれらを８個単位で結合していくことで署名情報が取り出せる.

得られた量子化テーブル（隠蔽量子化テーブル）

16　　11　　13　　15　　12　　11　　16　　14

0001 0000　0000 1011　0000 1101　0000 1111　0000 1100　0000 1011　0001 0000　0000 1110

0　　1　　1　　1　　0　　1　　0　　0

0111 0100

得られた隠蔽情報

図 4.5　署名情報抽出の様子

得られた 0111 0100 は ASCII で”t”となる．これを量子化テーブル２つ分行い，署名情報を抽出する.

第５章　ネットワーク監視モニタ

5.1　モニタ概要

モニタマシンはネットワークを監視するモニタプログラムを搭載する．モニタマシンはパケットレベルで常にサーバからの転送及び，サーバへの転送を監視し，機密文書の流出を検知する．検知した場合，その文書の重要度に応じた対応をとることができる．本研究では監視の対象を FTP と SMTP に絞りモニタプログラムを試作，その実用性について検証を行った．

5.2　モニタプログラムの機能

本研究で作成したモニタプログラムには次の機能がある．

・パケットキャプチャ機能
・パケットフィルタ機能
・プロトコル解析機能（FTP，SMTP）
・Word 文書ファイルから JPEG 画像を抽出する機能
・JPEG 画像の量子化テーブルを解析し，署名情報を抽出する機能
・メール送信機能
・FTP におけるファイル転送停止機能

これらの機能について説明する．

5.3　パケットキャプチャ

本研究では BSD ライセンスに基づいて配布されている WinPcap というパケットキャプチャドライバを用いて，パケットキャプチャを行っている．本モニタプログラムでは WinPcap3.0 を使用している．

WinPcap の機能としては

・ Ethernet レベルでネットワークを流れるすべてのパケットを取得できる
・ 自分宛てでないパケットを取得できる（無差別透過モード）
・ Ethernet レベルでの任意のパケットをネットワーク上に流すことができる
・ キャプチャまたは送信するパケットをフィルタすることができる

などが挙げられる．これらは本研究のモニタプログラムの

・ パケットキャプチャ機能
・ パケットフィルタ機能
・ パケット生成機能（FTP ファイル転送停止機能）

の実現に用いている．

WinPcap の詳細については参考文献[6]を参照されたい

5.4　モニタプログラムの動作

モニタプログラムの各モジュールの役割について説明する（図 5.1）．

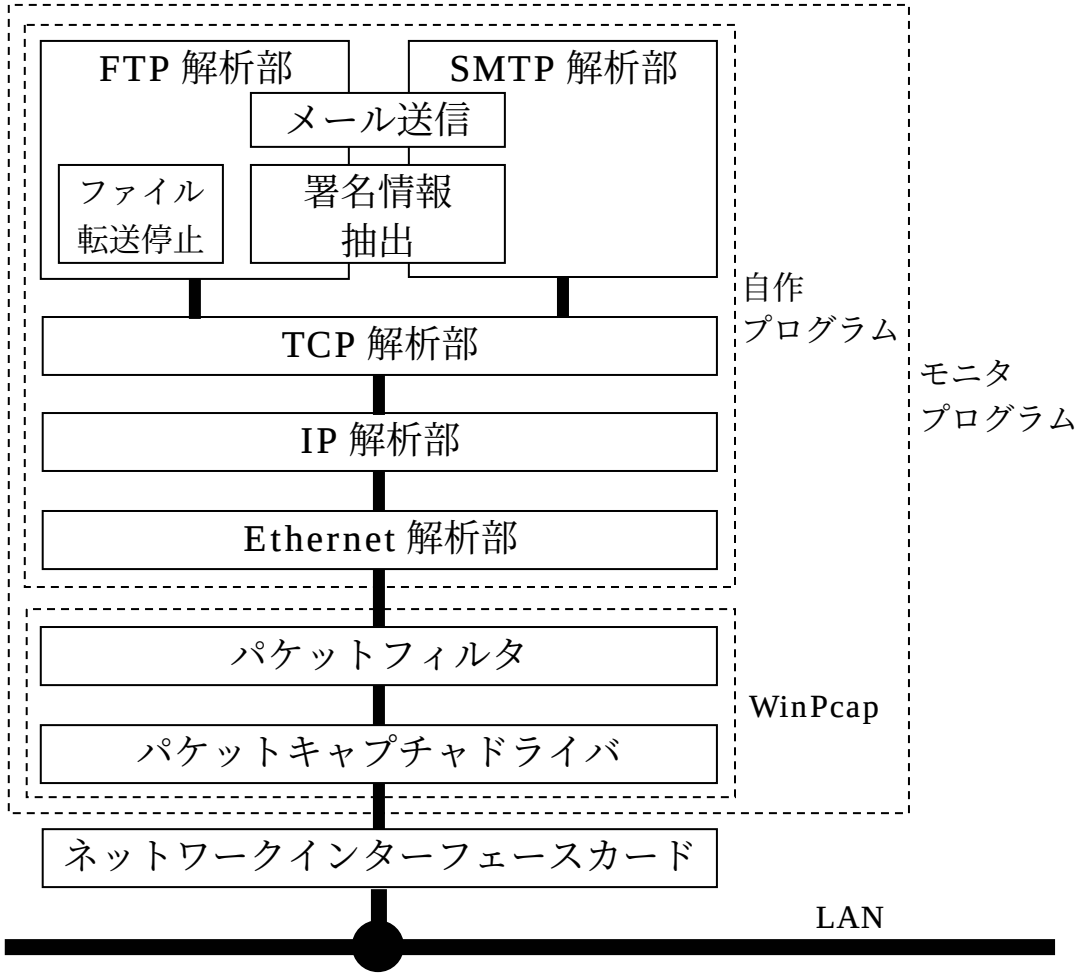


図 5.1　モニタプログラムの各モジュールの概観

ネットワークインターフェースカード(NIC)に届いたパケットは WinPcap のパケットキャプチャドライバを通してモニタプログラムに取り込まれる．本来自分宛で無いパケットは取得することはできないが，WinPcap では NIC を無差別透過モードにすることで自分宛でないパケットも取得できる．

取得されたパケットは WinPcap のパケットフィルタを通してから自作部分へと送られる．

パケットフィルタを通過したパケットは Ethernet，IP，TCP へとデマルチプレクスされる．その動作内容はヘッダ部分を解析し，そのパケット情報を保存し上位の関数に

引き渡している．

デマルチプレクスの結果，FTP か SMTP のパケットであったらそれぞれの解析部へと送られる．FTP と SMTP のモジュール内の解説はそれぞれの節で説明する．メール送信と署名情報抽出部は FTP，SMTP で共通の関数を用いている．

パケットと各プロトコルについては参考文献[7][8]または Ethernet は RFC894，IP は RFC791，TCP は RFC793 を参照されたい

5.5 FTP

5.5.1 FTP とは

FTP(File Transfer Protocol)はインターネットやイントラネットなどの TCP/IP ネットワークでファイルを転送するときに使われるプロトコルであり, 現在のインターネットで HTTP や SMTP／POP と並んで頻繁に利用されるプロトコルである．ファイルの漏洩に関し，まずファイル転送の基本ともいえるこのプロトコルへの対応を行った．

FTP に関しては参考文献[7][8]または RFC959 を参照されたい．

5.5.2 FTP 解析部

FTP 解析部の詳細は次のとおりである．

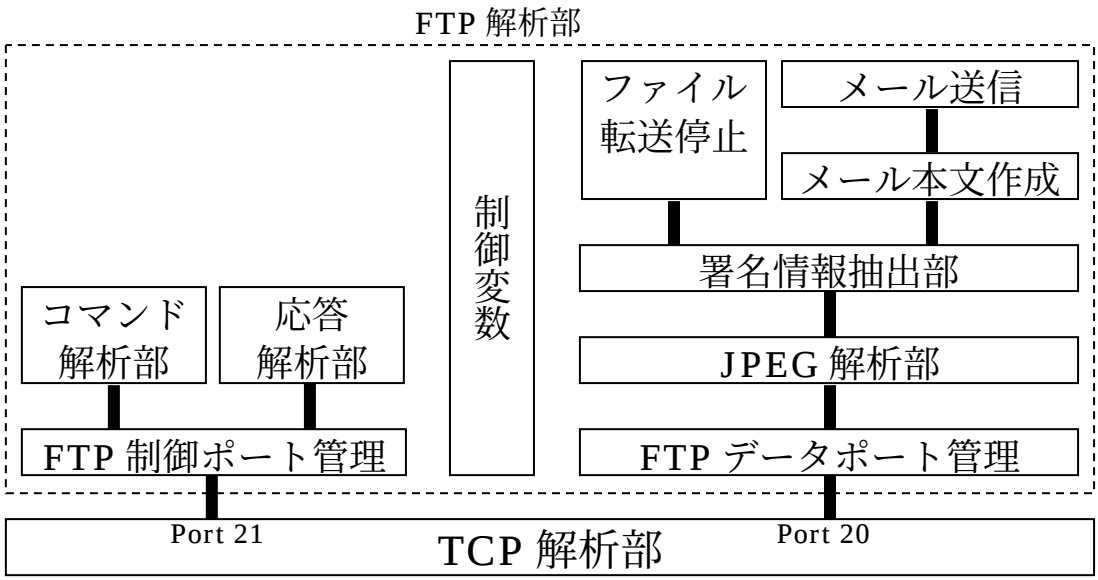


図 5.2　FTP 解析部

TCP 解析部にて解析された情報はポート別に FTP 制御ポート管理，または FTP データポート管理へと引き渡される．FTP 解析部の情報はすべて接続別に制御変数に格納される．それぞれの関数は制御変数から現在の状態を知る． JPEG 解析部，署名情

報抽出部，メール送信部は SMTP と共通の関数を用いる．

FTP では制御用とデータ用の２つのポートを使用する．制御用のポートはログインのためのユーザ名や，パスワード認証，ファイル転送に伴うコマンドの送信などに使われる．制御ポートは通常 21 番が使われる．一方，データ用のポートはデータの転送のみに使用され, ファイルの分割データとその確認応答, リストなどの転送に使用される．データポートは通常 20 番が使われる．
FTP を制御する FTP コマンドには次のような種類がある．

- アクセス制御用コマンド
- 転送パラメータ設定コマンド
- FTP サービスのためのコマンド

それぞれのコマンドの詳細は参考文献[8]を参照されたい．本モニタではこれらのコマンドのなかで，表 5.1 に挙げたコマンドのみ特別な対応をとるようになっている．これらのコマンドはクライアントアプリケーションからサーバアプリケーションへ送信されるコマンドで，実際にユーザがコンソールに打つコマンドとは異なる．

表 5.1　モニタプログラムが対応する FTP コマンド

コマンド	意味
USER ユーザ名	ユーザ名の入力
PASS パスワード	パスワードの入力
QUIT	正常終了
PORT h1,h2,h3,h4,p1,p2	データコネクションに使用する IP アドレスと ポート番号の指定
RETR ファイル名	FTP サーバからデータをダウンロードする

またコマンドに対してサーバアプリケーションはクライアントアプリケーションへ FTP 応答メッセージを送信する．FTP 応答メッセージは次のような種類がある．

- 情報の提供
- コネクション管理に関する応答
- 認証とアカウントに関する応答
- 不特定のエラー
- ファイルシステムに関する応答

それぞれの応答の詳細は参考文献[8]を参照されたい.
本モニタではこれらの応答のなかで, 表 5.2 に挙げた応答のみ特別な対応をとるようになっている.

表 5.2　モニタプログラムが対応する FTP 応答

応答番号	意味
220	Command okay.
221	Service closing control connection. Logged out if appropriate.
226	Closing data connection. Requested file action successful.
230	User logged in, proceed.
331	User name okay, need password.

FTP 制御ポート管理部は, これらのコマンド／応答から状態を制御する変数（表 5.3）を用いて接続を管理する.

表 5.3　FTP 制御状態

状態	意味
Phase 1	FTP コネクション
Phase 2	ユーザ名要求
Phase 3	パスワード要求
Phase 4	コマンド受付状態
Phase 5	ファイル転送状態(サーバ→クライアント)

FTP データポート管理部は解析時間計測，JPEG 解析部へパケット情報の引渡し，レベルごとの対応を管理する.

5.5.3　実際のモニタ動作

FTP による文書流出時のモニタの動作ついて実際の流れに沿って説明する．Phase については表 5.3 を参照されたい.

- コネクションからログインまで（図 5.3）

① SYN パケットを検知すると, その SYN パケットを送信したホストをクライアント, 送られたホストをサーバとして認識する．そして新たな接続としてパラメータを保存するメモリ領域を確保し，監視体制に入る.
② Phase2 になり，ユーザ名保存モードへ移行.
③ ユーザ名を保存する.
④ Phase3 になり，パスワード保存モードに移行.

⑤ パスワードを保存する．ただしこれは拡張機能であり，この情報を利用する機能は搭載していない．

⑥ Phase4 になり，コマンド解析モードに移行．

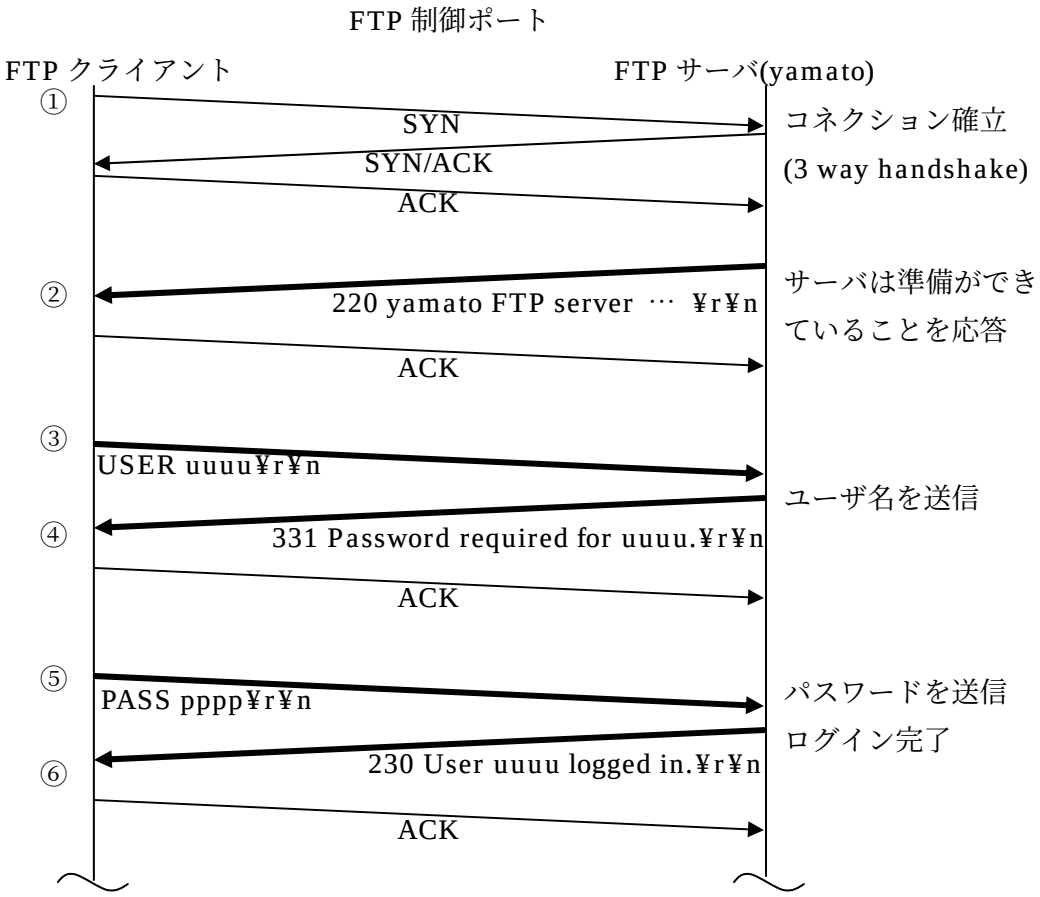


図 5.3　FTP コネクションからログインまで

- ファイル転送の準備（図 5.4）

⑦ データポート番号の保存を行う．

⑧ RETR コマンド（表 5.1）からファイル名を取得し，拡張子が doc の場合，Phase5 に移行し，データポートの解析を行う．

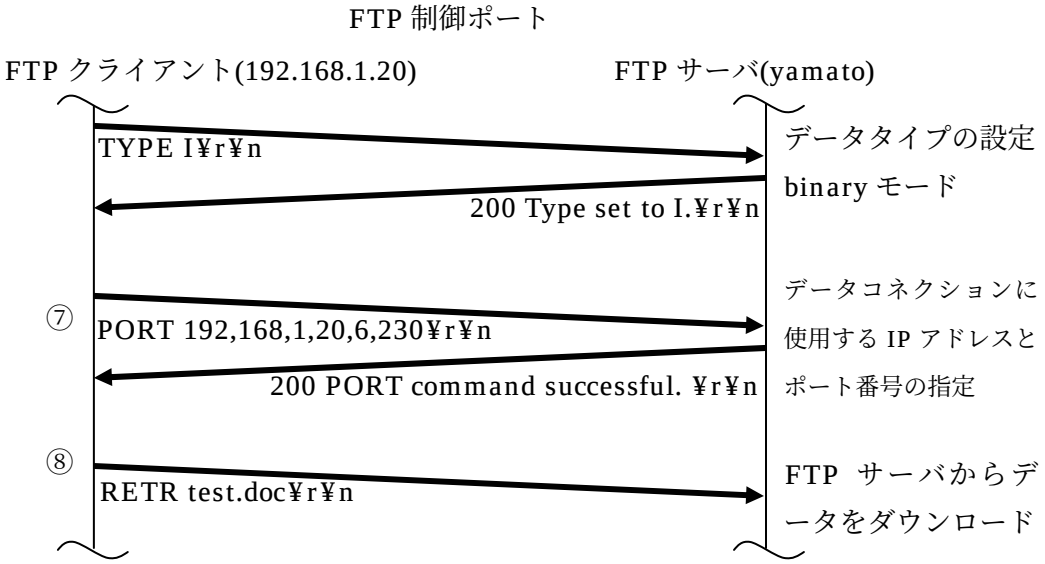


図 5.4　ファイル転送の準備

- データポートの準備（図 5.5）

⑨ RETR コマンドの後サーバはデータポートとしてクライアントに指定されたポートに対してコネクション要求を行う．ここから解析時間の計測を始める．

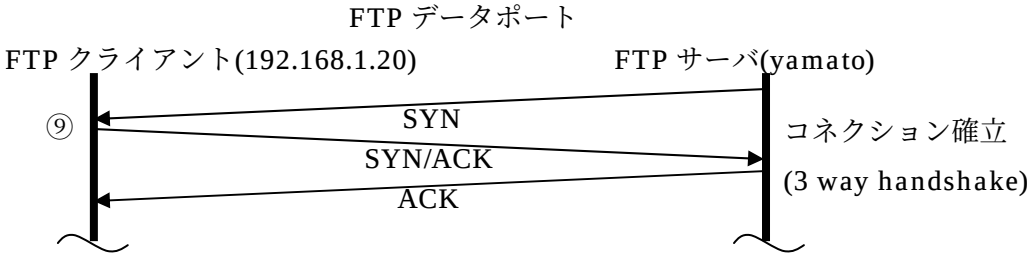


図 5.5　データポートの準備

- データポートコネクション確立の通知（図 5.6）

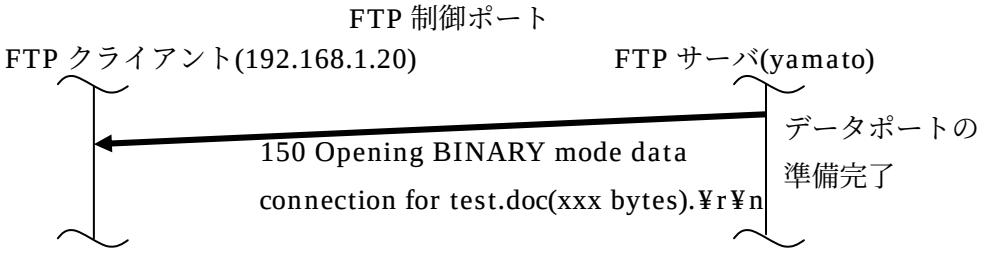


図 5.6　データポートコネクション確立の通知

- ファイルの転送（図 5.7）

⑪ サーバからのファイル転送を監視し，JPEG の検索および JPEG の量子化テーブルから署名情報抽出を行う．JPEG 画像は１つのみ解析され，量子化テーブルから署名情報が抽出される，されないにかかわらず，データポートの監視を終了し，制御ポートの監視に戻る．また時間の計測を終了する．署名情報よりファイルが重要度レベル３の文書であった場合は直ちにファイル転送の停止を行う．その後，今までに収集した情報を管理者，役員などにメールで送信する．このメールアドレスはあらかじめコンフィグファイルにて指定しておく．Phase4 に戻り，コマンド監視モードになる．

⑫ 画像が抽出されなかった場合，解析が打ち切られないためここで Phase4 に戻り，時間の計測を終了する．

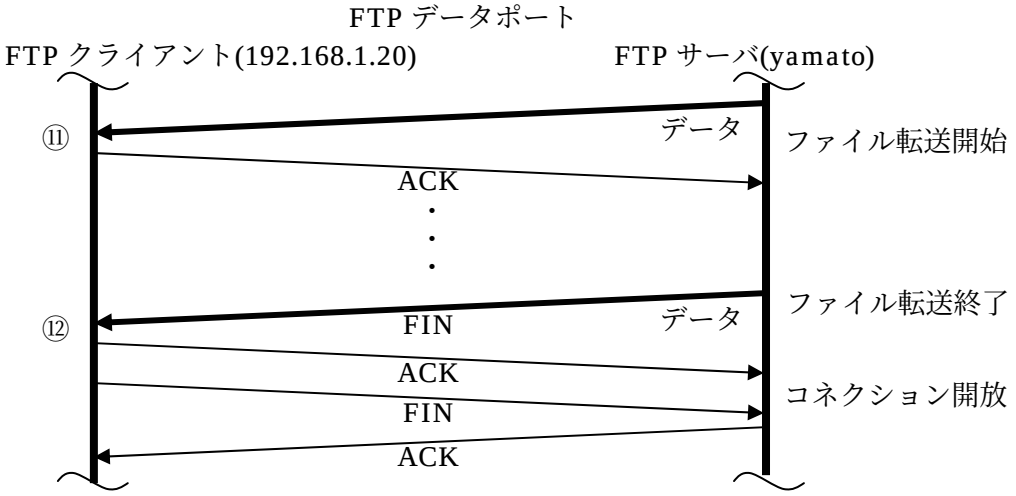


図 5.7　ファイルの転送

- ファイル転送の終了（図 5.8）

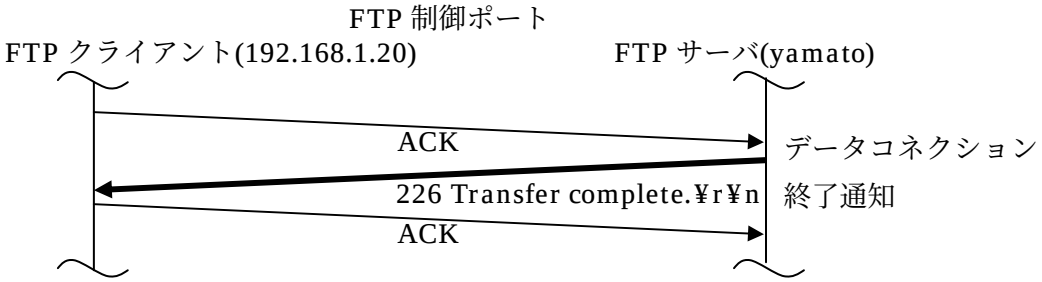


図 5.8　ファイル転送の終了

- FTP の終了（図 5.9）

⑬ サーバからの 221 応答メッセージによって，このコネクションの監視を終了し，メモリを開放する．

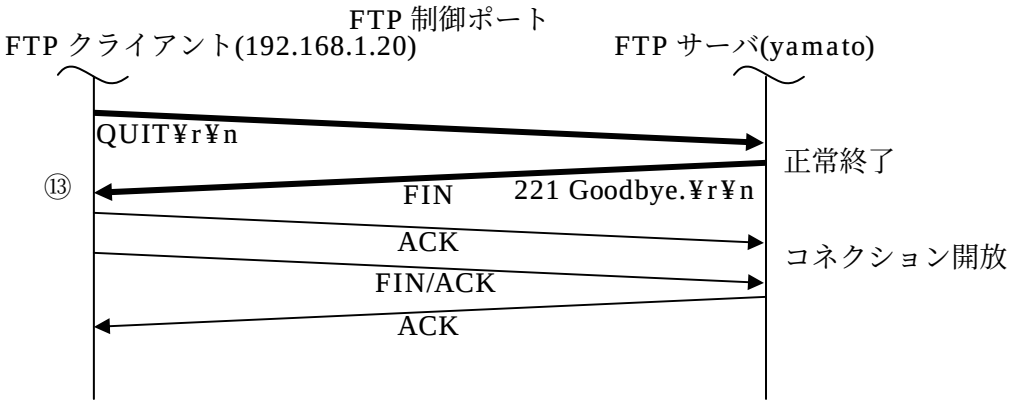


図 5.9　FTP の終了

5.5.4　FTP ファイル転送停止

サーバからクライアントへのファイル転送を中止するにはクライアント側が送信の中断コマンドを送信する必要がある．しかしモニタはサーバでもクライアントでもない第３者であるため，モニタがファイル転送を停止するには特殊な方法が必要となる．

そこで本システムでは WinPcap の機能の一つである任意のパケットを送信する機能を用いて，クライアントがサーバへファイル転送停止コマンドを送信する際のパケットを偽装して送信することで停止することを試みた．

本来クライアントがサーバに対してファイル転送の停止を行う場合，データポートではなく制御ポートを使用して次のようなデータを送信する必要がある．

```
<IAC, IP, IAC, DM, A, B, O, R, \r, \n>
  ( IAC : Interpret As Command
    IP  : Interrupt Process
    DM  : Data Mark )
```

このデータ送信には２つのパケットを用い，最初のパケットは緊急ポインタを指定して送信する必要がある．このパケットがクライアントから送信されたものであると偽装するため，ヘッダの MAC アドレス，IP アドレス，ポート番号，シーケンス番号，ACK 番号をそれまでに監視していたデータから偽装する．これによってサーバはクライアントからファイル転送の停止要求があったものと誤解し，ファイルの転送を停止する．実際に送信

したパケット次のようになる.

```
2003-09-17 22:45:21
パケット長              : 43 bytes
送信元ポート番号        : 3382
送信先ポート番号        : 21
シーケンス番号          : 48549689
ACK 番号                : 3752754366
ヘッダ長                : 20 bytes
フラグ                  : PUSH ACK URG
緊急ポインタ            : 00 03
## data ##
FF F4 FF                : <IAC, IP, IAC>

2003-09-17 22:45:21
パケット長              : 47 bytes
送信元ポート番号        : 3382
送信先ポート番号        : 21
シーケンス番号          : 48549692      ←   48549689 + 3
ACK 番号                : 3752754367    ← 3752754366 + 1
フラグ                  : PUSH ACK
## data ##
F2 41 42 4F 52 0D 0A    : <DM, A, B, O, R, CR, CF>
```

しかしこの方法にはいくつかの問題点がある.

・ コマンド受理されるまでに送信された内容はクライアントに残る
・ サーバの NIC のキューに入っているパケットは送信を停止することはできない
・ 検知したパケットが最終パケットであった場合，転送を停止することができない
・ パケットの送信までに時間がかかった場合，転送の停止が間に合わないことがある

これらの解決方法としては，IP マスカレードを用いてモニタし，故意に遅延を作ることでファイル転送を停止することが可能ではないかと考えられる.

5.6 SMTP

5.6.1 SMTP とは

SMTP(Simple Mail Transfer Protocol)は電子メールを送信するためのプロトコルである．サーバ間でメールのやり取りと，クライアントがサーバにメールを送信する際に用いられる．そして機密文書が漏洩する際にもっとも使われるプロトコルであると思われる．本システムではこの SMTP に対する検知機能を実装した．

5.6.2 SMTP 解析部

SMTP 解析部の詳細について解説する．

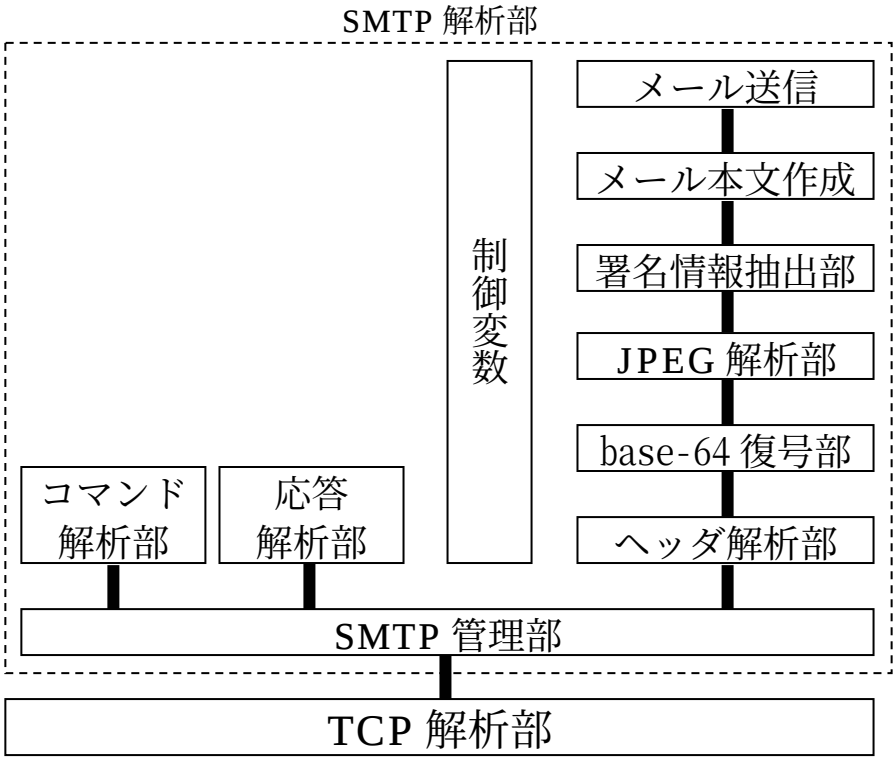


図 5.10　SMTP 解析部

SMTP ではポートは一つなので TCP 解析部にて解析された情報はすべて SMTP 管理部へと引き渡される．FTP と同様，SMTP 解析部の情報はすべて接続別に制御変数に格納される．それぞれの関数は制御変数から現在の状態を知る．SMTP では制御情報とデータが同じポートで流れるため管理は複雑になっている．またメールの本文はヘッダとメッセージと添付が含まれるため，この制御も複雑である．添付は base-64 にて符号化されているため，添付データは base-64 復号部を通してから JPEG 解析部へと送られる．SMTP 解析部の関数のうち JPEG 解析部，署名情報抽出部，メール送信部は FTP と共通の関数を用いる．

SMTPではFTPと異なり１つのポート（ポート25)でコマンドとデータを両方扱う．そのため処理は FTP より複雑となる．またメールヘッダの解析及びメールの構造の把握が必要となる．そしてもっとも重要なことは添付が符号化されていることである．つまり機密文書が符号化されているため JPEG 検索するためにはそれを復号する必要があることである．

本モニタでは SMTP コマンドの中で，表 5.4 に挙げたコマンドのみ特別な対応をとるようになっている．SMTP コマンドの詳細については参考文献[8]または RFC821 を参照されたい．

表 5.4　モニタプログラムが対応する SMTP コマンド

コマンド	意味
HELO <domain>	通信開始
MAIL FROM:<reverse-path>	送信者
RCPT TO:<forward-path>	受信者の指定(Receipt to)
DATA	電子メールの本文の送信

またコマンドに対してサーバアプリケーションはクライアントアプリケーションへSMTP応答メッセージを送信する. SMTP 応答メッセージの種類は次のとおりである．

- 要求に対する肯定確認応答
- データの入力
- 転送エラーメッセージ
- 処理の継続が不可能なエラー

SMTP 応答の詳細については参考文献[8]または RFC821 を参照されたい．本モニタではこれらの応答のなかで, 表5.5に挙げた応答のみ特別な対応をとるようになっている．

表 5.5　モニタプログラムが対応する SMTP 応答

応答	意味
220 <domain>	サービスを開始する
221 <domain>	サービスを終了する
354	電子メールのデータの入力開始．ピリオド(.）だけの行で入力終了

電子メールは次の３つの部分で構成される．

- エンベロープ
- ヘッダ
- 本文

エンベロープ（封筒）は，メールを送信するために MTA（メッセージ転送エージェント）によって用いられる．SMTP コマンドの MAIL FROM:<reverse-path>，RCPT TO:<forward-path>はエンベロープの指定にあたる. エンベロープについては RFC821 を参照されたい.

ヘッダは，ユーザ・エージェントによって用いられる．本モニタは表 5.6 に挙げるヘッダ・フィールドのみ対応している．その他のヘッダ・フィールドについては RFC822 を参照されたい.

表 5.6　モニタプログラムが対応するメール・ヘッダ・フィールド

ヘッダ・フィールド	意味
Content-Type	文字タイプと文字コードセットを指定
Content-Transfer-Encoding	文書符号化方法指定

ユーザ・エージェントであれば Received, From, To, Reply-to などに対応する必要があるが, 本システムはモニタであり必要な情報はエンベロープから取得しているためヘッダ・フィールドの解析は最低限のものにしている.

本文は，送り手のユーザから受けてのユーザへ送られるメッセージの内容である．RFC822 によって本文は NVT ASCII テキストと定められている．DATA コマンドを用いて転送されるとき, まずヘッダが送られ, 次に空行が続き, その後に本文が送られる.

添付ファイルは多目的 Internet メール拡張機能(MINE)を用いている．これによって ASCII 以外のデータも送信可能となる．上記に上げたヘッダ・フィールドは MINE の拡張ヘッダ・フィールドである．添付ファイルは通常 base-64 変換されマルチパートとして本文内にある．MIME については RFC1522 を参照されたい．添付ファイルを取り出すにはこれらを基にメールの構造について解析する必要がある.

SMTP 管理部ではこれらのことを FTP と同様，制御変数（表 5.7）を用いて制御している.

表 5.7　SMTP 制御状態

状態	意味
Phase 1	SMTP コネクション
Phase 2	コマンド受付
Phase 3	ヘッダ／本文解析

そして，MINE のマルチパートを制御するための変数（表 5.8）を使用する.

表 5.7　マルチパート制御状態

状態	意味
-1	打ち切り
0	初期状態
1	マルチパート解析中
2	マルチパートの添付部分(base-64 復号中)
3	マルチパートの添付以外の部分
4	ファイル名取得中

5.6.3　実際のモニタ動作

SMTP による文書検出の流れについて実際の流れに沿って説明する．Phase に関しては表 5.7 を参照されたい.

- コネクションからコマンドの受付まで（図 5.11）

① FTP と同様, SYN パケットを送信したホストをクライアント, 送られたホストをサーバとして認識する．そして新たな接続としてパラメータを保存するメモリ領域を確保し，監視体制に入る.

② 220 応答メッセージ(表 5.5)をもって Phase2 に移行し，コマンドの解析を始める.

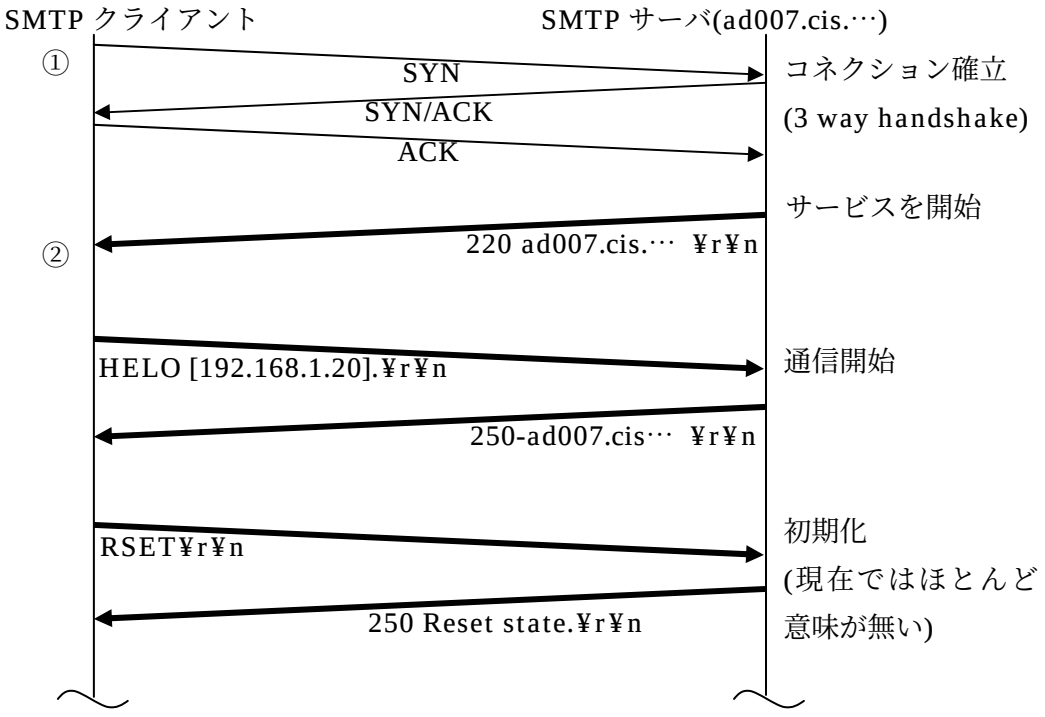


図 5.11　コネクションからコマンドの受付まで

● エンベロープの送信（図 5.12）

③ MAIL コマンドより送信元アドレスを取得する.

④ RCPT コマンドより送信先アドレスを取得する.

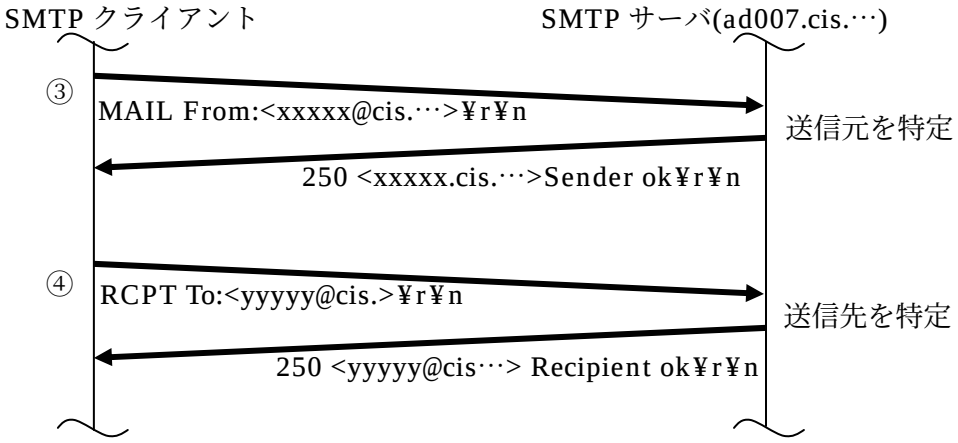


図 5.12　エンベロープの送信

● 本文の送信（図 5.13）

⑤ 354 応答メッセージをもって時間計測を開始する. Phase3 に移行し, ヘッダ, 本文, 添付の解析を行う. 本文解析の流れは後に示す. 添付内から画像が抽出された場合, 時間を計測し，”.”が出現するまで本文の解析をしない.

⑥本文の解析, 時間の計測を終了する. Phase2 に移行し, コマンド解析モードに戻る.

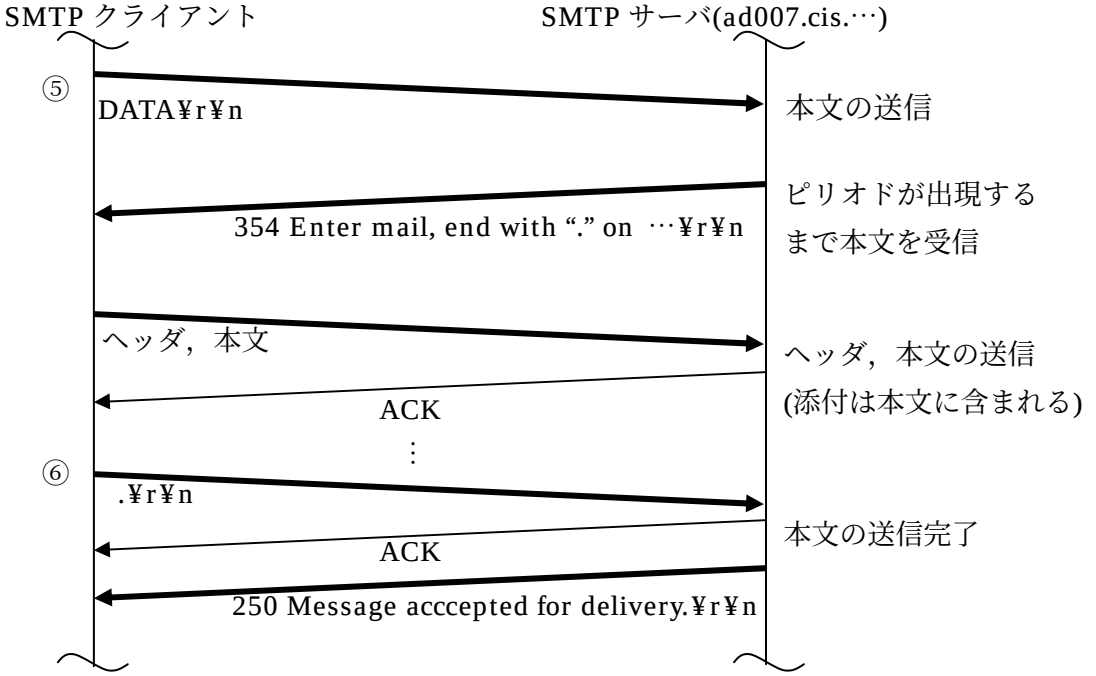


図 5.13　本文の送信

- SMTP の終了（図 5.14）

⑦ 221 応答メッセージをもって監視を終了し，メモリを開放する．

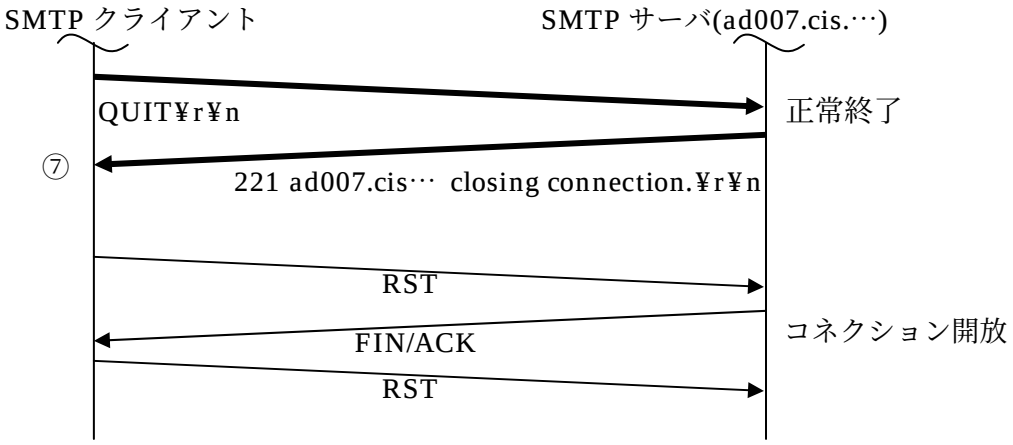


図 5.14　SMTP の終了

次に Phase3 のメールヘッダ／本文／添付の解析について説明する．

① ヘッダ・フィールドの Content-Type が multipart/mixed; であることを確認する．multipart/mixed;の場合，マルチパートの境界を表す boundary 属性が指定されているのでこれを保存する(制御変数 multipart = 1 となる)．　multipart/mixed; でない場合，本文の解析をしない(multipart = -1).

② 境界が出現すると，そのパートの種類が Content-Type ヘッダ・フィールドで指定される．Word 文書ファイルの場合は application/octet-stream;となる(multipart = 2)．これ以外の場合は，そのパートの解析を行わない(multipart = 3).

③ Word 文書ファイルの場合，ファイル名が指定される．本システムではまだ日本語に対応していないため，ファイル名の保存は可能だが，拡張子によってファイルを判別することはできていない(multipart = 4 ファイル名の保存が終了すると 2 に戻る).

④ そして Content-Transfer-Encoding ヘッダ・フィールドによって添付のエンコード方法が指定される．本システムでは最も一般的な base-64 符号化のみ対応している．

⑤ Content-Transfer-Encoding ヘッダ・フィールドのあとは符号化された添付が続く．これを base-64 復号部で復号し，JPEG 解析部へと送る．画像が抽出された場合，本文の解析を終了する(multipart = -1).

⑥ 画像が抽出されなかった場合，”.”の出現をもってメール本文の解析を終了する(Phase2(表 5.7)に戻る).

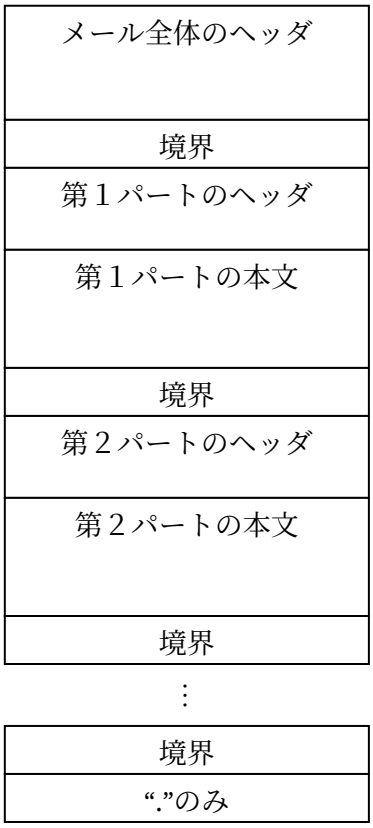


図 5.15　マルチパートのメール構造

5.7　メール送信機能

文書の流出が検知された場合，どの重要度においてもメールを送信して，検出を伝えるようにしている．送信先のアドレスはあらかじめコンフィグファイルにて設定されており，メール送信時に読み込むようにしてある．

この機能は SMTP を使ったメール送信プログラムであるが，その設計時間が無く日本語を送信する機能を持ち合わせていない．ファイルの拡張子による打ち切りのためにも，今後日本語に対応させる必要がある．

5.7.1　FTP による文書流出時のメール内容

FTP の場合のメール送信内容は以下のようになる．

1. 文書流出検出時間　　検出時のパケットより取得
2. ユーザ ID　　コマンドから取得
3. クライアント MAC アドレス　　コネクション開始時のパケットより取得
4. クライアント IP アドレス　　コネクション開始時のパケットより取得
5. 転送ファイル名　　コマンドから取得

6. 文書の重要度

実際のメール本文

FTP leak detect
2004-02-15 18.31.20(JST)
userID xxxx
MAC address 00 E0 4C 81 02 82
IP 192.168.1.20
transfered_file 1M1logo.doc
level 3

5.7.2 SMTP による文書流出時のメール内容

FTP の場合のメール送信内容は以下のようになる.

1. 文書流出検出時間　　検出時のパケットより取得
2. クライアント MAC アドレス　　コネクション開始時のパケットより取得
3. クライアント IP アドレス　　コネクション開始時のパケットより取得
4. 送信元アドレス　　エンベロープより取得
5. 送信先アドレス　　エンベロープより取得
6. 転送ファイル名　　メール本文から取得
7. 文書の重要度

実際のメール本文

SMTP leak detect
2004-02-15 11.09.44(JST)
MAC address 00 E0 4C 81 02 82
IP 192.168.1.20
FROM <xxxxx@cis.shimane-u.ac.jp>
TO <yyyyy@cis.shimane-u.ac.jp>
TO <zzzzz@cis.shimane-u.ac.jp>
filename 1M1logo.doc
ALERT LEVEL 3

第６章　実験

本研究で作成したシステムの性能を測定するため，文書流出の検知実験を行った.

6.1　実験環境

・FTP サーバ

OS　　: Red Hat Linux 7.0.1J
CPU　　: K6-Ⅱ 300MHz
RAM　　: 256MB
FTP　　: Version wu-2.6.1(1)

・SMTP サーバ（学科内 SMTP サーバ）

ESMTP Sendmail 8.9.3

・クライアントマシン

OS　　: WindowsXP Pro SP1
CPU　　: Celeron 1.80GHz
RAM　　: 256MB

・モニタマシン

OS　　: WindowsXP Pro SP1
CPU　　: AthlonXP 1500+(1.34GHz)
RAM　　: 256MB
WinPcap 3.0

・FTP ソフト

FFFTP ver 1.91

・SMTP ユーザ・エージェント(メールソフト)

Becky! Internet Mail version 2.05.04

・ネットワーク

100Base-TX

・文書作成

Microsoft Word 2002 SP2

・ソフトウェア開発環境

Microsoft Visual Studio 6.0

6.2　FTP による文書流出実験

・計測点

(1)画像を含む文書が流出した場合

パケットキャプチャ開始
　FTP 接続開始
　　ファイル転送開始　　←　計測開始
　　ロゴ画像検出
　　署名情報抽出　　←　署名情報検出終了時間
　　ファイル転送停止　　←　ファイル転送停止コマンド送信終了時間
　　メール送信
　　ファイル転送終了　　←　ファイル転送終了時間（打ち切り無し）
　FTP 接続開放

(2)画像を含まない文書が流出した場合

パケットキャプチャ開始
　FTP 接続開始
　　ファイル転送開始　　←　計測開始
　　ファイル転送終了　　←　ファイル転送終了時間
　FTP 接続開放

・ネットワーク構成図

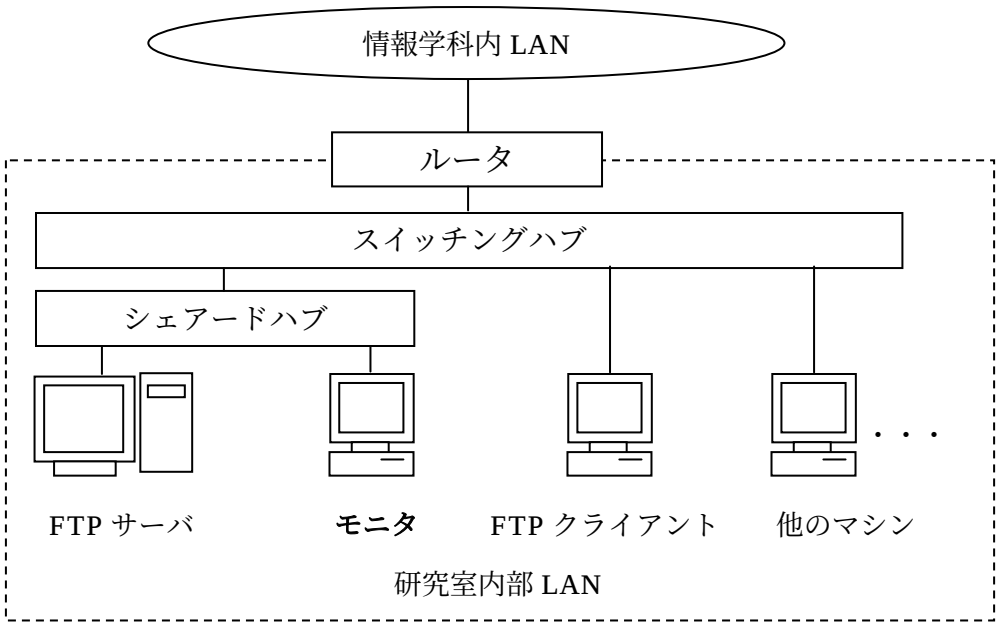


図 6.1　FTP による文書流出の監視の際の接続

● 実験１　ロゴ画像の有無による変化

実験条件

・ ロゴ画像：サイズ 4.46KB　隠蔽文字 top secret
・ 文字数を変化させることでファイルサイズを変更
・ 実験は 10 回行い，その平均を求める

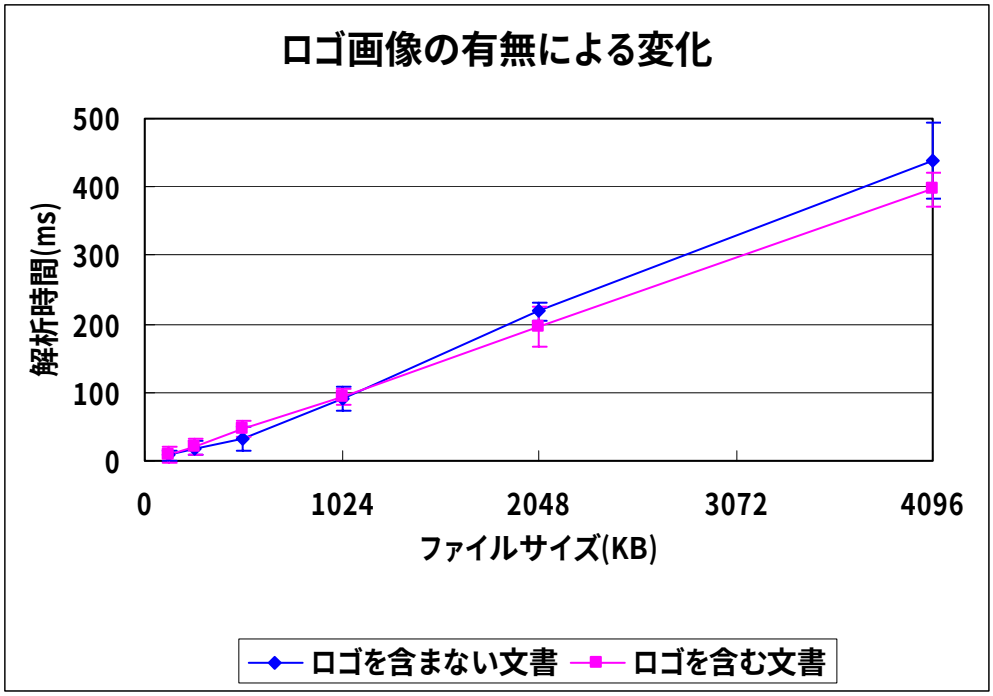
実験結果１

ロゴ画像を含まない文書

サイズ(KB)	128	256	512	1024	2048	4096
平均解析時間(ms)	7.8	18.7	32.8	90.6	218.2	438.8
分散	67.7	99.1	292.0	321.6	177.1	3092.4
標準偏差	8.2	10.0	17.1	17.9	13.3	55.6

ロゴ画像を含む文書

サイズ(KB)	128KB	256KB	512KB	1024KB	2048KB	4096KB
平均解析時間(ms)	9.5	20.4	45.5	93.8	195.2	396.8
分散	123.8	167.4	136.3	163.7	875.5	607.1
標準偏差	11.1	12.9	11.7	12.8	29.6	24.6



（各点は平均を表し，上下の線は標準偏差を表す.）

考察

ロゴ画像を含むものはロゴ画像から署名情報が抽出された段階で, 処理が打ち切られる．そのため画像のイメージデータおよび文書のフッタが解析されないためである．また文書の構造から文字が画像より先に保存されているため, 解析時間は文字数に比例する．

- 実験２　画像の追加による変化

実験条件

・ 実験１の 1MB のロゴ画像を含む文書に 200K の JPEG 画像を等間隔に追加する
・ 文字数は変化させない
・ 対照実験として，モニタを１つ目の画像で解析打ち切りしないように設定し，本来ファイルの転送にどれだけかかるかを測定する
・ 実験は 10 回行い，その平均を求める

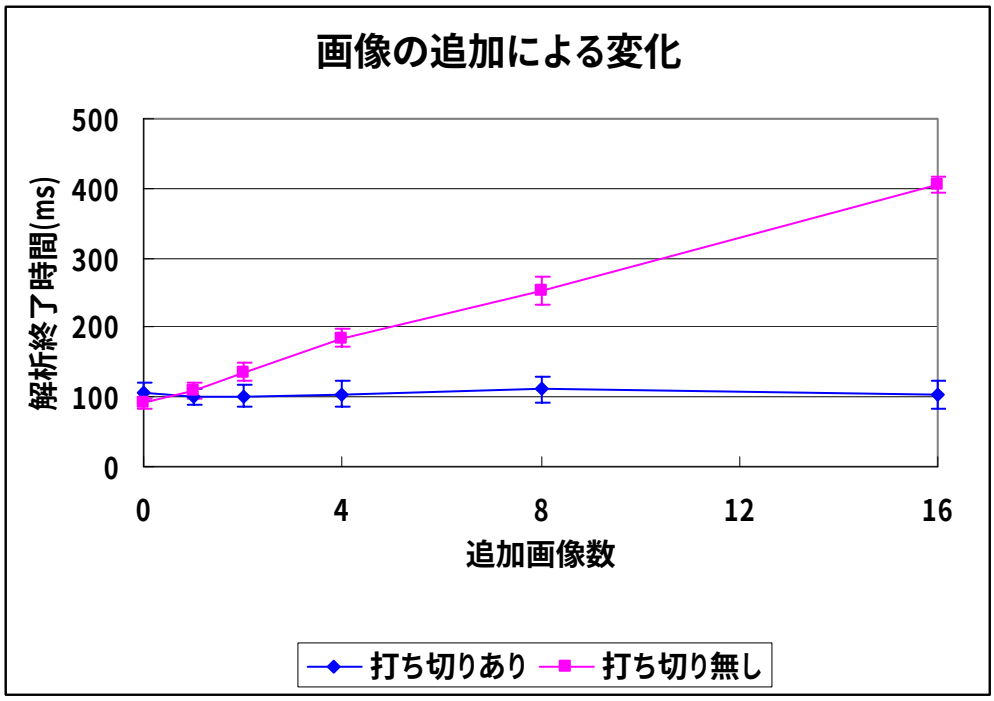
実験結果２

打ち切りあり

追加画像数	0	1	2	4	8	16
ファイルサイズ(KB)	1024	1258	1466	1880	2709	4366
平均解析時間(ms)	107.7	101.4	101.7	104.8	110.9	103.1
分散	186.9	124.7	279.6	329.5	345.0	395.7
標準偏差	13.7	11.2	16.7	18.2	18.6	19.9

打ち切り無し

追加画像数	0	1	2	4	8	16
ファイルサイズ(KB)	1024	1258	1466	1880	2709	4366
平均解析時間(ms)	90.6	109.2	136	184.2	253.2	406.4
分散	49.8	120.2	155	166.7	435.2	128.3
標準偏差	7.1	11.0	12.4	12.9	20.9	11.3



考察

本システムはファイルの最初に出現する JPEG 画像しか解析しない．つまり最初に保存されているロゴ画像しか解析されないので, 画像を追加してもその処理時間は変化しない. 対照実験の打ち切り無しのグラフと比較すると画像数が多いほど処理時間が軽減できていることがわかる.

● 実験３ FTP ファイル転送停止にかかる時間

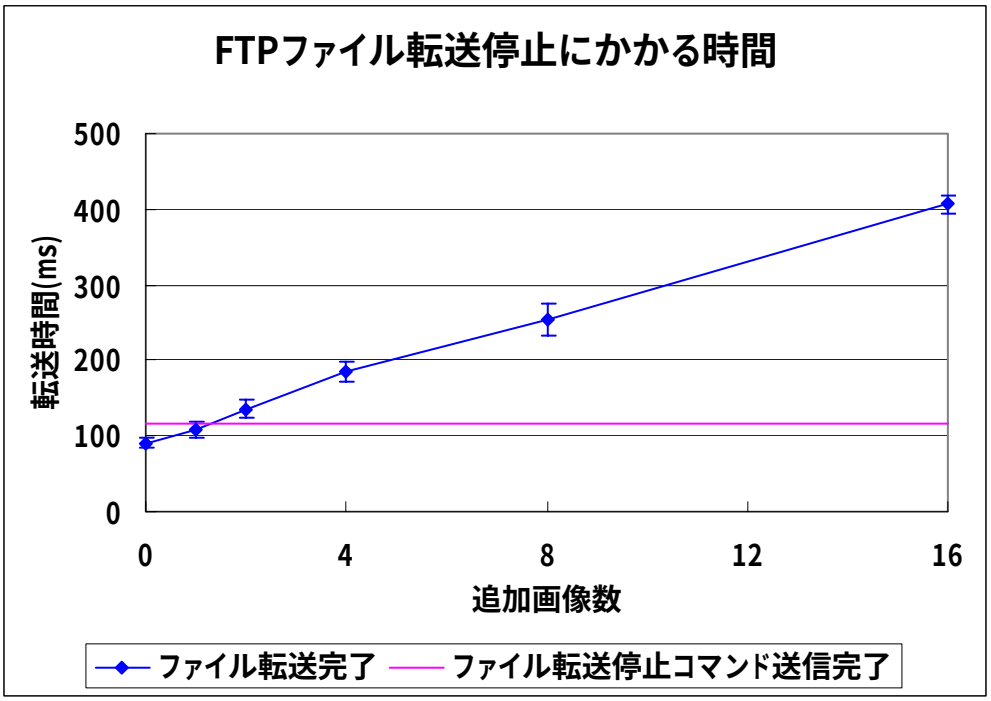
実験条件

- 署名情報を検出してから FTP ファイル転送停止コマンド送信完了までにかかった時間を計測
- 全計測結果の平均を取る
- 実験２の対照実験と比較する

実験結果３

ファイル転送停止コマンドを送信するまでにかかった時間

0.0245 秒



考察

追加画像数が少ない文書ではファイル転送停止コマンドを送信しても間に合わない可能性がある．追加画像が８枚以上であれば停止コマンドが有効であると予測できる．

しかし署名情報が検出された段階で既に文書部分が流出しているため完全に停止するには別の方法が必要となる．

6.3　SMTP による文書流出実験

・計測点

パケットキャプチャ開始

SMTP 接続

本文送信開始　　←　計測開始

ロゴ画像検出

署名情報抽出　　←　署名情報検出終了時間（打ち切りあり）

メール送信

SMTP 接続開放　　←　ファイル転送終了時間（打ち切り無し）

・ネットワーク構成

本来は SMTP サーバとシェアードハブで並列接続するべきであるが，今回は研究室内で計測するため SMTP クライアントと並列接続した（図 6.2）.

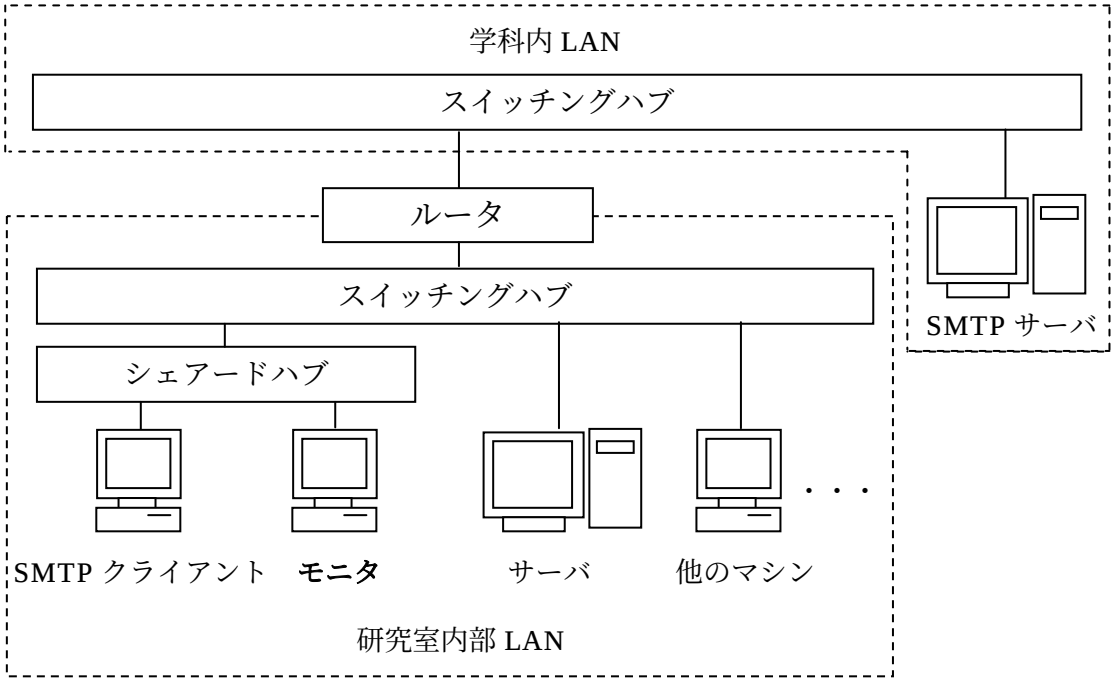


図 6.2　FTP による文書流出の監視の際の接続

● 実験４　ロゴ画像の有無による変化

実験条件

・ 送信先は１箇所で CC, BCC は無い
・ メール本文は英文で 100 文字
・ 添付は実験１と同様のサンプルにて計測

実験結果４

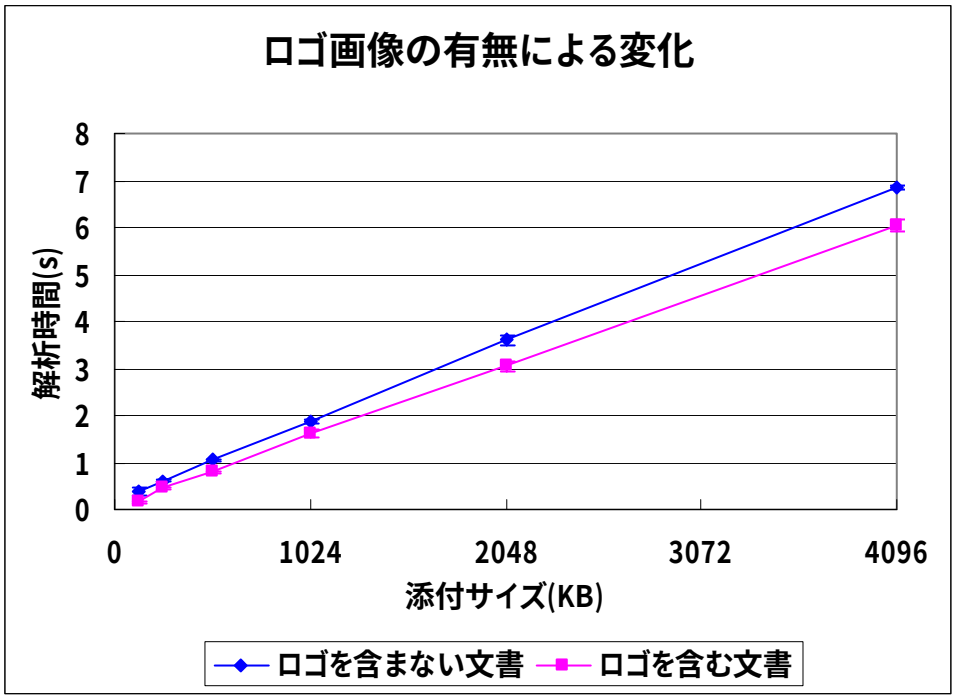
ロゴを含まない文書

添付サイズ(KB)	128	256	512	1024	2048	4096
メールサイズ(KB)	176	351	763	1411	2817	5611
平均解析時間(s)	0.3954	0.6094	1.0453	1.8781	3.6014	6.8594
分散	0.735E-2	0.085E-2	0.156E-2	0.305E-2	1.106E-2	0.211E-2
標準偏差	0.0857	0.0292	0.0394	0.0552	0.1052	0.0459

ロゴ画像を含む文書

添付サイズ(KB)	128	256	512	1024	2048	4096
メールサイズ(KB)	176	351	763	1411	2817	5611
平均解析時間(s)	0.1624	0.4502	0.7906	1.6220	3.0566	6.0438
分散	0.031E-2	0.017E-2	0.019E-2	0.515E-2	0.999E-2	2.079E-2
標準偏差	0.0176	0.0129	0.0138	0.0718	0.1000	0.1441

メールサイズはヘッダと本文と添付ファイルを含むサイズ. 添付部分は base-64 変換にてサイズが 4/3 倍になっている.



考察

FTP と同様に文字数に比例する結果が得られた. FTP に比べて解析時間が大きいのはメールソフトが送信に時間がかかっていることと, 情報分野 SMTP サーバを用いたため研究室内よりトラフィックが多いことによる遅延が想定される. そのためロゴ画像を含む画像のほうが, 早く解析終了することが FTP の場合よりよくわかる.

- 実験５　画像の追加による変化

実験条件

・ 送信先は１箇所で CC, BCC は無い
・ メール本文は英文で 100 文字

・ 添付は実験２と同様のサンプルにて計測

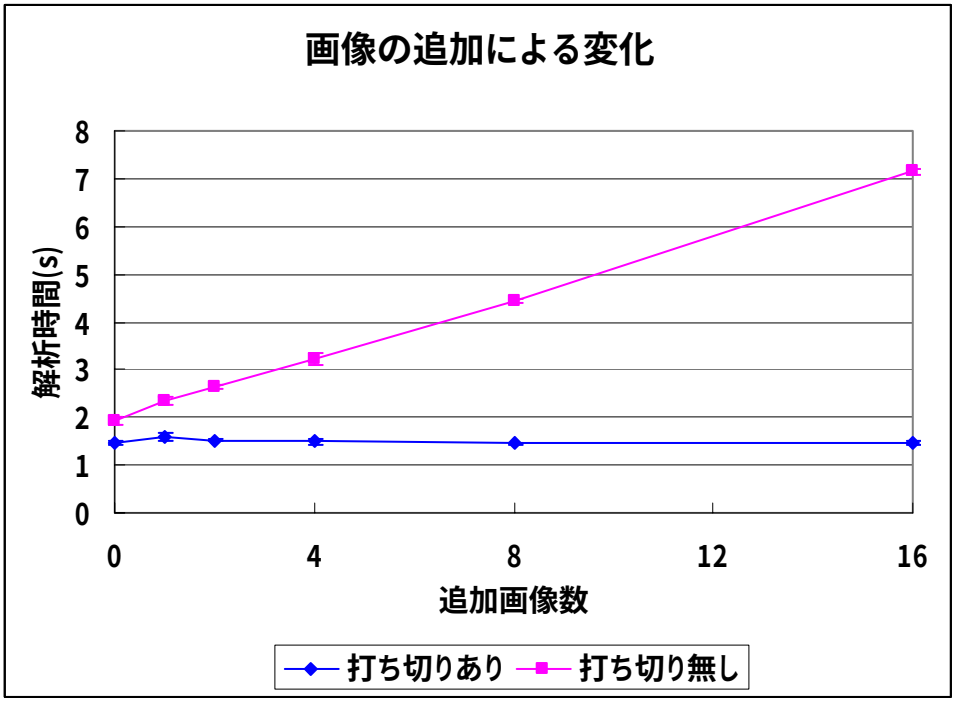
実験結果５

打ち切りあり

追加画像数	0	1	2	4	8	16
添付サイズ(KB)	1024	1258	1466	1880	2709	4366
メールサイズ(KB)	1445	1723	2008	2574	3709	5976
平均解析時間(s)	1.4796	1.5719	1.5215	1.4890	1.4468	1.4750
分散	0.150E-2	0.709E-2	0.078E-2	0.364E-2	0.142E-2	0.099E-2
標準偏差	0.0388	0.0842	0.0280	0.0603	0.0377	0.0314

打ち切り無し

追加画像数	0	1	2	4	8	16
添付サイズ(KB)	1024	1258	1466	1880	2709	4366
メールサイズ(KB)	1445	1723	2008	2574	3709	5976
平均解析時間(s)	1.9297	2.3532	2.6327	3.2174	4.4342	7.1547
分散	0.505E-2	0.561E-2	0.247E-2	1.387E-2	0.168E-2	0.324E-2
標準偏差	0.0710	0.0749	0.0497	0.1178	0.0410	0.0569



考察

FTP の場合と同様に，打ち切りによる解析時間の軽減が確認できた．この実験においても FTP の 10 倍程度の処理時間がかかったが，前の実験で述べたとおり，SMTP サーバまたはユーザ・エージェントがボトルネックになっていると思われる．

第 7 章　終論

7.1　まとめ

本研究をとおして定型文書にステガノグラフィを用いて署名情報を隠蔽したロゴ画像を挿入することで, 利用者に認知されずに署名情報を付加し, パケットキャプチャ機能を持ったモニタプログラムを用いてリアルタイムにネットワークを監視し, パケットレベルに分割された文書ファイルのロゴ画像から署名情報を抽出することで, 専用の機器を使用せずに文書流出の検出を行うことができた.

また本システムではロゴ画像に署名情報を隠蔽したことにより, 文書内からロゴ画像のみを検索するのみで機密情報か否かを判別でき, 全文検索システムに比べ, 検索にかかる負荷を大幅に軽減することができた.

追加機能であるが, FTP において署名情報抽出以後のデータに関しては転送の停止ができた. またメールの送信を行い, 文書の流出を伝えることができるようになった.

本研究を通してネットワークの低い層のから上位の層の代表的なプロトコルまで深く学ぶことができた.

7.2　今後の展望と課題

現在のシステムはファイルが圧縮された場合対応できない. 現実の世界においてメールで送信する際は圧縮するのが当たり前となっている. そのため圧縮ファイルに対する対応は必須のものとなる. またファイル名が偽装されている場合を考え, そのファイルのヘッダ情報からファイルを特定する必要もあるかもしれない.

本システムはリアルタイム時に容易に署名情報を抽出できるよう, 簡単なステガノグラフィを適用している. 簡単なステガノグラフィは抽出にかかる時間も少なくて済むが, 実際の運用を考えるとステガノグラフィの高度化が必要になると考えられる. また本システムは１度付加されたロゴ画像の署名情報を簡単に変更することができない. その点を含めて新たな隠蔽方法を模索する必要がある.

また追加機能として搭載した FTP のファイル転送停止機能であるが, 完全にファイルを停止することはできなかった. 実際にファイル転送停止を行うのであれば, 完全に停止できる機能を搭載する必要がある.

今回の実験はクライアントが１台の場合しか計測できなかった. そのためモニタの性能を十分に測定できたとはいい難い. 今回行った実験のほかに, 複数のクライアントからサーバアクセスさせたり, モニタマシンの性能を変えて実験したりすることで, リアルタイムシステムとしての性能を評価する必要がある.

謝辞

本研究にあたり，最後まで熱心な御指導をいただきました田中章司郎教授には，心より御礼申し上げます．また，田中研究室の皆さんには，本研究に関して数々の御協力と御助言をいただきました．厚く御礼申し上げます．

なお，本論文，本研究で作成したプログラム及びデータ，並びに関連する発表資料等のすべての知的財産権を，本研究の指導教官である田中章司郎教授に譲渡致します．

参考文献・資料

[1] 松井甲子雄：電子透かしの基礎 -マルチメディアのニュープロテクト技術-，森北出版，1998
[2] 松井甲子雄：画像深層暗号 -手法と応用-，森北出版，1998
[3] 小野束：電子透かしとコンテンツ保護，オーム社，2001
[4] 小野文孝，渡辺裕：国際標準符号化の基礎技術，コロナ社，1998
[5] 原島博 他：画像情報圧縮，オーム社，1991
[6] WinPcap：http://winpcap.polito.it/
[7] W・リチャード・スティーヴンス 著，橘康雄 訳：詳解 TCP/IP Vol.1 [新装版] プロトコル，ピアソン・エデュケーション，2000
[8] 松下隆史，村山公保，荒井透，苅田幸雄：マスタリング TCP/IP 入門編 第２版，オーム社，1998

付録１　JPEG マーカ

- JPEG のファイル構成

JPEG はマーカと呼ばれる FFXX で表される２バイトの印によってその情報を区別する．またマーカとそのパラメータを合わせてセグメントと呼ぶ．マーカとその意味は次のとおりである．

表　マーカとその意味

マーカ	シンボル	意味
FF00		SOS 中でのデータ FF
FF01	TEM	算術符号の個人使用データ
FF02-BF	RES	予備
FFC0	SOF0	フレーム開始：ベースライン DCT，ハフマン
FFC1	SOF1	フレーム開始：拡張シーケンシャル DCT，ハフマン
FFC2	SOF2	フレーム開始：プログレッシブ DCT，ハフマン
FFC3	SOF3	フレーム開始：Spatial DPCM，ハフマン
FFC4	DHT	ハフマンテーブルの定義
FFC5	SOF5	フレーム開始：差分シーケンシャル DCT，ハフマン
FFC6	SOF6	フレーム開始：差分プログレッシブ DCT，ハフマン
FFC7	SOF7	フレーム開始：差分 Spatial DPCM，ハフマン
FFC8	JPG	予備
FFC9	SOF9	フレーム開始：拡張シーケンシャル DCT，算術符号
FFCA	SOF10	フレーム開始：プログレッシブ DCT，算術符号
FFCB	SOF11	フレーム開始：Spatial DPCM，ハフマン
FFCC	DAC	算術符号の状態の定義
FFCD	SOF13	フレーム開始：差分シーケンシャル DCT，算術符号
FFCE	SOF14	フレーム開始：差分プログレッシブ DCT，算術符号
FFCF	SOF15	フレーム開始：差分 Spatial DPCM，算術符号
FFD0-D7	RSTn	再開
FFD8	SOI	イメージ開始(Start Of Image)
FFD9	EOI	イメージ終了(End Of Image)
FFDA	SOS	スキャン開始　符号化されたイメージデータ開始
FFDB	DQT	量子化テーブルの定義
FFDC	DNL	ライン数
FFDD	DRI	再開間隔の定義
FFDE	DHP	ハイアラーキカル方式

FFDF	EXP	拡張参照ページ
FFE0-ED	APPn	アプリケーション用予備
FFF0-FD	JPGn	JPEG 用予備
FFFE	COM	コメント

● マーカの構成

この中で通常使われるものについて出現順に詳しく説明する.

<SOI>,<EOI>はマーカのみから構成され，それ以外のマーカは情報の長さとマーカ情報から構成されている.

表　マーカの構成

マーカ	シンボル	説明・パラメータ
FFD8	SOI	イメージ開始(Start Of Image) JPEG ファイルの先頭にあり，マーカのみで構成される. (2byte)マーカ = FFD8
FFE0	APP0	アプリケーションデータセグメント JFIF で使用する解像度情報など (2byte)マーカ = FFE0 (2byte)マーカサイズ = 16(10) (4byte) “JFIF” (1byte)00 (2byte)Format Version = 0101 (1byte)単位 = 00 (2byte)水平解像度 (2byte)垂直解像度 (1byte)サムネイル画像横サイズ (1byte)サムネイル画像縦サイズ
FFE1-ED	APPx	アプリケーションデータセグメント 例えば Adobe PhotoShop 7 では FFE1 を使っている. (2byte)マーカ = FFEx (2byte)マーカサイズ (可変長)データ
FFDB	DQT	量子化テーブル定義セグメント 複数のテーブルを１つのマーカで書くことも，複数のマーカに分けて書くこともできる. (2bytes)マーカ = FFDB (2bytes)Lq(量子化テーブル定義長)

		{ (4bit)Pq(量子化テーブル制度) = 0:8bit, 1:16bit (4bit)Tq(量子化テーブル番号) = 0～3 (64 or 128byte)Qn(データ 64 個) }*t
FFC4	DHT	ハフマンテーブル定義セグメント 複数のテーブルを１つのマーカで書くことも，複数のマーカに分けて書くこともできる. (2byte)マーカ = FFC4 (2byte)Lh(ハフマンテーブル定義長) { (1byte)Th(ハフマンテーブル識別子) 上位 4bit = 0:DC 成分, 1:AC 成分 下位 4bit = 識別番号 0～3 (1byte)Li(長さ i の符号数) (1byte)Vij(それぞれの符号語に関連する値) }*t
FFC0	SOF0	フレームヘッダ：標準 DCT 圧縮 (2byte)マーカ = FFC0 (2byte)Lf(フレームヘッダ長) (1byte)P(サンプル制度) = 08 (2byte)Y(垂直サイズ) = 0～65535 (2byte)X(水平サイズ) = 0～65535 (1byte)Nf(フレーム内の画像成分数) = 1:グレースケール 3:カラー { (1byte)Ci(成分識別子) (4bit) Hi(水平サンプリングファクタ) (4bit) Vi(垂直サンプリングファクタ) (1byte)Tqi(量子化テーブルセレクタ) }*i(成分数)
		スキャン開始(Start Of Scan) 以降符号化されたイメージデータが続く. (2byte)マーカ = FFDA (2byte)Ls(SOS のヘッダ長) (1byte)Ns(スキャン内の画像成分数) = 1:グレースケール

		3:カラー
FFDA	SOS	{ (1byte)Csi(スキャン成分セレクタ) (4bit)Tdi(DC 用ハフマンテーブルセレクタ) (4bit)Tai(AC 用ハフマンテーブルセレクタ) }*i (1byte)Ss(ベースライン DCT では未使用) = 0 (1byte)Sc(ベースライン DCT では未使用) = 63 (4bit)Ah(ベースライン DCT では未使用) = 0 (4bit)Al(ベースライン DCT では未使用) = 0
FF00		SOS 中でのデータ FF データ FF とマーカ FFXX と区別するためにある
FFD9	EOI	ファイルの終端

付録２　モニタプログラムのオプション

● モニタプログラム detective のオプション指定

・オンラインモード

detective [-o output_file] [-p "packet_filter"][-F][-S][-d]

NIC よりパケットを取得するモード.

-o オプションが指定された場合，-F -S -d オプションは無効となる.

・オフラインモード

detective -f input_file [-o output_file] [-p "packet_filter"][-F][-S][-d]

-f オプションによって指定されるファイルからの読み込みを行うモード.

-o オプションが指定された場合，-F -S -d オプションは無効となる.

-f filename	**オフラインモードにおける入力ファイル指定 このオプションを指定した場合，NIC からパケットの取得を行わず，代わりに指定されたファイルのパケット情報を読み込む**
-o filename	**出力ファイル指定 このオプションが指定された場合，指定されたファイルにパケット情報を書き出す**
-p "packet_filter"	**パケットフィルタ設定**
-F	**FTP 監視モードを有効にする**
-S	**SMTP 監視モードを有効にする**
-d	**デバッグモード（隠しオプション）**

● キャプチャフィルタの主な指定方法

詳しくは http://winpcap.polito.it/docs/man/html/group__language.html

・基本フィルタ

（*host*：名前または IP アドレス，*ehost*：イーサーネットアドレス）

フィルタ	意味
dst host *host*	**あて先 IP アドレス**
src host *host*	**送信元 IP アドレス**
host *host*	**送信元または宛先 IP アドレス**
ether dst *ehost*	**ethernet あて先アドレス**
ether src *ehost*	**ethernet 送信元アドレス**

ether host *ehost*	ethernet 送信元または宛先アドレス
ip proto *protocol*	IP パケットのプロトコル icmp，udp，tcp などを指定可能
icmp，udp，tcp	ip proto *protocol* 短縮形
ether proto *protocol*	ethernet のプロトコル ip，arp，rarp などが指定可能
ip，arp，rarp	ether proto *protocol* の短縮形

・等値演算子

(*x*，*y* ：基本フィルタ)

x == *y*	*x* と *y* は等しい
x != *y*	*x* と *y* は等しくない

・関係演算子

x > *y*	*x* は *y* より大きい
x < *y*	*x* は *y* より小さい
x >= *y*	*x* は *y* 以上である
x <= *y*	*x* は *y* 以下である

・論理演算子

not *x* ! *x*	否定
x and *y* *x* && *y*	論理和
x or *y* *x* || *y*	論理積

・算術演算子

( )	括弧

- フィルタ例

"4(port 20 or 21) and host 192.168.1.1"

"tcp port 25 and host 10.210.40.7"