Vol.2011-DPS-146 No.11
Vol.2011-CSEC-52 No.11
2011/3/10

HTTP 通信の秘匿のためのステガノグラフィを用いたプロキシの開発

犬飼辰夫[†]　　平野学[†]

インターネットの検閲システムによる表現の自由の問題が表面化する一方，非合法的な情報の流通を監視することを目的としたシステムの確立も求められている．このような状況において秘匿通信方式とその検出方法についての研究の重要性が増しつつある．本稿では秘匿通信方式の一つとしてステガノグラフィ技術を用いたウェブ通信の秘匿通信システムを提案した．提案方式は監視システムに画像の送受信を行っているようにみせかけながら，実際は別のウェブサイトとの通信を行うものである．本稿ではプロトタイプ実装の性能評価を行った結果を報告した．最後に提案方式の検出方法の議論をおこなった．

A Steganography-based HTTP Proxy System for Covert Web Communications

Tatsuo Inukai[†]　and Manabu Hirano[†]

In these days, some countries deploy censorship mechanisms of the Internet. On the other hand, some countries try to deploy efficient surveillance systems for antisocial and illegal contents. Therefore, studies of such censorship and surveillance systems are becoming more important. This paper shows a novel steganography-based HTTP proxy system for covert web communications. The proposed system bypasses surveillance systems on the Internet. The proposed system can embed original HTTP messages into image files by steganography and send them to original destination servers secretly. This paper shows results of performance evaluation of the prototype implementation. This paper discusses the detection method of the proposed system.

1. はじめに

インターネットが社会基盤としての重要性を増すにつれて，通信を介しての犯罪行為や非合法的なコンテンツのやりとりを取り締まる監視システムの重要性が増しつつある．しかしながら技術的に同じシステムを用いると，インターネットを流れるデータの検閲が可能になり政治的な思想を取り締まる道具としても利用される可能性がある[1]．この種の監視システムは一般市民にとって有益にも働くことがあれば，必ずしもそうではない場合もある．しかしながら監視システムに関する研究の重要性が増しつつあることは確かである．

本稿では実際に利用されている監視システムの技術的な分類を示す．次に監視システムを迂回する技術について説明する．本稿では監視システムを迂回するシステムの一つとしてステガノグラフィ技術を用いて通信の存在を隠してウェブ通信を行なうシステムの設計を示す．提案システムはステガノグラフィ技術で監視システムに対して有意な通信が存在することを秘匿する．本稿では提案システムのプロトタイプ実装と性能評価の結果を示す．最後にこの種の技術は悪用される可能性もあることから監視システムでの効率的な検知方法について考察する．

2. 監視システムの分類

本稿ではまずインターネット通信の監視システムについて技術的な分類を行う．本稿で監視システムとはある組織によって不都合な情報を監視し，場合によっては該当する通信を遮断する，あるいは該当する通信の記録を残しておくことができるシステムを意味するものとする．

2.1 ブラックリスト方式

接続を禁止したいホストの IP アドレスの一覧を事前にブラックリストとして登録しておき，ルータでフィルタリングする方式である[2]．ブラックリスト方式の利点はフィルタリングの実装が簡単なことである．ファイアウォール製品に実装されている場合には特別なハードウェアを追加せずにそのまま利用できる．ブラックリスト方式の一つとして DNS タンパリングと呼ばれる方法もある．DNS タンパリングでは接続を禁止したいホストのドメインネームの一覧を事前にブラックリストとして登録しておき，DNS への問い合わせ時に本来の IP アドレスではない偽の IP アドレスを返すものである．例えば「この接続先の閲覧は認められていません」といった情報を表示するホストの IP アドレスを返すことができる．

2.2 ディープパケットインスペクション

ブラックリスト方式の根本的な問題はリストを維持管理することが困難なことである．さらにホストの一部のコンテンツだけが有害である場合に，同一ホストのすべての通信を遮断することが問題になることもあり得る．近年の国家レベルの検閲などで用いられているのがディープパケットインスペクション（Deep Packet Inspection, DPI）である．DPI は監視地点を通過するパケットを解析し，コンテンツに禁止されたキーワードが含まれるかを検出する[3]．IXP などで莫大な量の通信を監視するためには専用のハードウェア装置が用いられる[1]．

2.3 プロキシソフトウェアとキーワード検出

企業などの組織レベルで従業員の通信を監視する用途にはアプリケーション層で

[†]豊田工業高等専門学校　専攻科
Toyota National College of Technology.

Vol.2011-DPS-146 No.11
Vol.2011-CSEC-52 No.11
2011/3/10

動作する簡易的なプロキシソフトウェアを用いることが多い．例えばウェブ通信を監視するには従業員のウェブブラウザからの通信を必ず組織とインターネットの境界に設置したフィルタリング用のプロキシサーバを経由させるようにすればよい．管理者がキーワードを設定して通信を遮断できる製品が広く利用されている．電子メールの監視機能が統合されたメールサーバも利用されている．どの方式も組織と外部の境界に設置されたサーバによってアプリケーションレベルでの監視を行うシステムである．

2.4 クライアントコンピュータで動作する監視プログラム

これまで述べた方法はどれも通信経路，特に境界に設置したサーバで監視を行うシステムであった．国によっては自国のすべてのコンピュータに監視プログラムのインストールを義務づけようという動きもあった[2]．あらかじめ全てのコンピュータに監視用プログラムのインストールを義務付け有害な画像やキーワードを検出して検閲を行う方法である．通信が集中する境界地点のサーバで監視するのに比べると画像解析による検査のように比較的負荷の高い処理を用いることができるのが利点である．ただし現実的に全てのコンピュータに監視用プログラムを導入することを強制させることが困難なためほとんど普及していない方式である．

3. 迂回システムの分類

監視システムを迂回するための技術について説明する．

3.1 暗号化通信と中継サーバを組み合わせた方式

監視システムを迂回するために外部のサーバを経由して通信を行う方式がある．例えば Anonymizer [4]では TLS 通信によって一旦外部サーバとの接続を行い，実際の目的ホストとの通信データは TLS による暗号化通信路を流すことで監視システムからのコンテンツ解析を防ぐものである．このような迂回用の中継サーバはブラックリスト方式で対応できることが多い．類似のサービスに GTunnel がある[5]．GTunnel はユーザのコンピュータで動作する HTTP/SOCKS5 プロキシソフトウェアが専用のサーバファームと暗号化通信することで匿名通信を実現する．同様に JonDonym [6]もクライアントで動作するプロキシソフトウェアであるが，GTunnel と比べるとサーバファームで通信を複雑にミックスして匿名性を高める工夫がなされている点が異なる．

3.2 通信経路の秘匿方式

監視システムの迂回には監視地点のサーバがコンテンツを見られないように通信内容を暗号化する方式が一般的である．しかしながらこの方式は中継サーバの一覧を登録したブラックリストを用意しておけば検出することができる．一方，暗号化とは別に通信経路を隠す方式がある．この方式を用いると特定の中継サーバをブラックリストに登録しているだけでは禁止したいサーバへの通信を検出することができない．このような通信経路の秘匿化を行う技術の一つにオニオンルーティングがある[7][8]．オニオンルーティングは複数の Tor サーバを用いて通信ごとに経路を変更することでトラフィック解析に対する耐性を高めている．接続先のホストの IP アドレスを隠せるほか接続元のホストの IP アドレスも隠せるためブラックリスト方式の監視システムを迂回できる．オニオンルーティングは米国の Naval Research Laboratory で開発されたものであり，各国の政府組織やジャーナリストが海外のウェブサイトを閲覧する際などにも利用されている[9]．

4. 既存システムの問題点

監視システムにはブラックリスト方式とコンテンツ検査方式があることを述べた．迂回システムには中継サーバを用いて監視拠点を暗号化通信で通過させる方式とオニオンルーティングのような通信経路の秘匿化を行う方式があることを述べた．迂回方式として暗号化通信を用いる方式には第三者から隠したい重要な情報の存在を示唆してしまう欠点がある．また通信経路の秘匿化を行う方式はブラックリストによる監視を迂回することができるが，オニオンルーティングの通信自体を遮断することは可能である．通信経路の秘匿化を行う際にもやはり重要な通信の存在を示唆してしまう欠点がある．そこで本稿では本来の通信の存在を隠しながらウェブサイトにアクセスするシステムの一例を示す．本稿で提案する方式は既存の監視システムを改良して検出することが可能であるが，コンテンツフィルタリングに要する処理を複雑にする．

5. ステガノグラフィを用いたウェブの秘匿通信方式

本稿では既存のウェブブラウザやサーバソフトウェアに変更を加えずに，秘密のウェブ通信を実現する方式を示す．図 1 にステガノグラフィ[10]を用いた秘匿通信方式を示す．提案方式では通信の存在を隠すために，あらかじめ監視地点の外側に 1 つ以上のサーバマシンを用意し，それらのサーバマシンでサーバ側プロキシを動作させておく必要がある．そして利用者のコンピュータあるいは監視システムより内側のコンピュータのどちらかにクライアント側プロキシを動作させておく．クライアント側プロキシとサーバ側プロキシの間で HTTP[11]のメッセージをステガノグラフィで画像に埋め込むことで，有意な情報の存在を知られることなく監視システムを通過させる．ウェブブラウザとクライアント側プロキシの間の通信と，ウェブサーバとサーバ側プロキシの間の通信がそれぞれ監視されていないことが前提となる．外部に 1 箇所以上のサーバマシンを用意して通信をランダムに分散させることでトラフィック解析による通信パターンの不自然さの検出をしづらくする．本稿では以上で述べた監視を迂回するシステムが仮に利用された場合に，どのように監視システムを改良して検出すればよいかを議論することを目的としてプロトタイプを開発した．

Vol.2011-DPS-146 No.11
Vol.2011-CSEC-52 No.11
2011/3/10

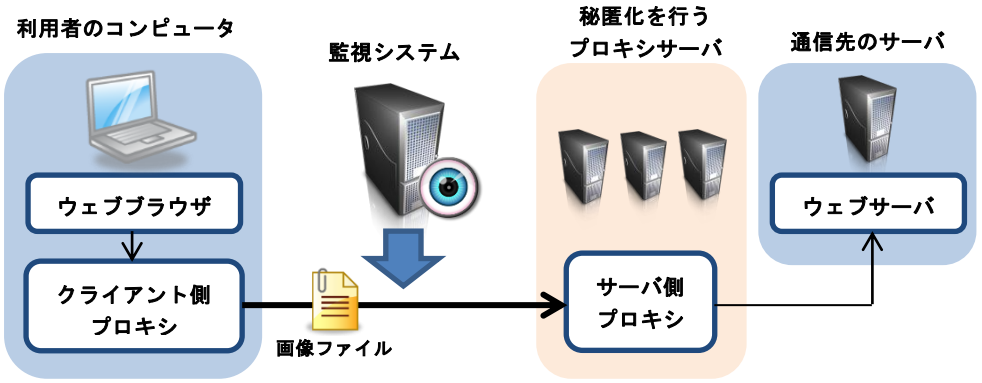

図 1　ステガノグラフィを用いた秘匿通信方式

6. ステガノグラフィを用いた秘匿通信プロキシの設計

クライアント側プロキシはウェブブラウザと直接やり取りを行うプロキシである．ウェブブラウザから受け取った HTTP リクエストを画像に埋め込みサーバ側プロキシに送信する処理，サーバ側プロキシから受け取った画像からレスポンスを抽出しウェブブラウザに送信する処理を行う．サーバ側プロキシはウェブサーバと直接やり取りを行うプロキシである．サーバ側プロキシはクライアント側プロキシから受信した画像からリクエストを取り出しウェブサーバへ送信する処理，ウェブサーバから HTTP レスポンスを受け取り画像に埋め込みクライアント側プロキシに返す処理を行う．

提案システムの HTTP 通信の流れを図 2 に示す．まず処理（1）でウェブブラウザが HTTP リクエストをクライアント側プロキシに送信する．次に処理（2）でクライアント側プロキシは HTTP リクエストを画像フォルダからランダムに選んだ画像ファイルにステガノグラフィで埋め込んでサーバ側プロキシに送信する．処理 (3) では HTTP リクエストの入った画像を受け取ったサーバ側プロキシは画像から HTTP リクエストを復元し，本来の目的地であるウェブサーバに送信する．処理（4）でウェブサーバは HTTP レスポンスをサーバ側プロキシに返す．処理（5）でサーバ側プロキシは HTTP レスポンスを再び画像フォルダからランダムに選んだ画像ファイルにステガノグラフィで埋め込んでクライアント側プロキシに返信する．処理（6）で HTTP レスポンスの入った画像を受け取ったクライアント側プロキシは最終的に画像から HTTP レスポンスを取り出して利用者のウェブブラウザに送信する．処理（2）と処理（5）で HTTP のリクエストとレスポンスを画像ファイルにステガノグラフィを用いて埋め込んでいる．ステガノグラフィ処理では埋め込まれるオブジェクトはテキストデータかバイナリデータかは気にしない．つまりウェブサーバからの HTTP レスポンスが画像や動画を含んでいる場合でも全て同じように図 2 の処理で対応できる．よって提案方式ではステガノグラフィの埋め込みが可能な大きさのオブジェクトならその形式に関わらず閲覧することが可能である．

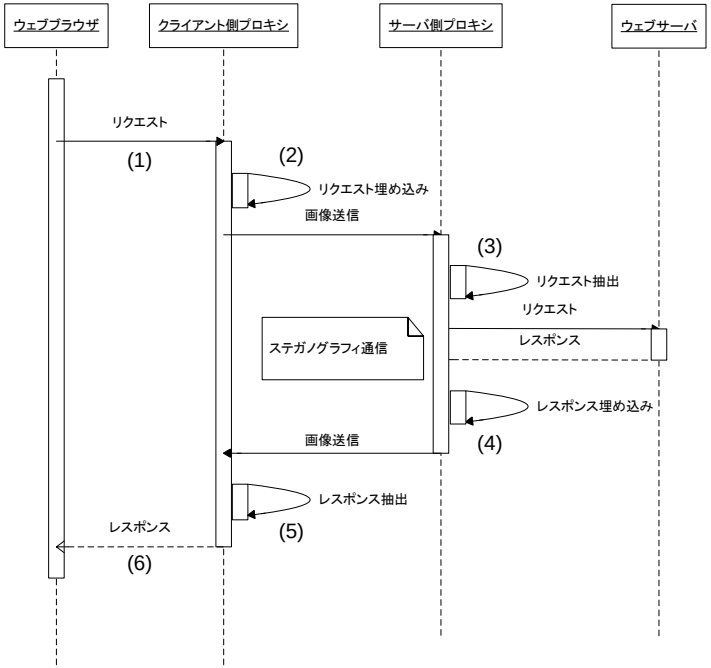

図 2　提案システムの通信の流れ

7. プロトタイプ実装

本稿では提案するステガノグラフィによる秘匿通信方式のプロトタイプを実装した．

Vol.2011-DPS-146 No.11
Vol.2011-CSEC-52 No.11
2011/3/10

プロトタイプ実装では HEAD メソッドと GET メソッドをサポートした．現時点では持続的接続（KeepAlive）はサポートしていない．クライアント側プロキシは Python の BaseHTTPServer モジュールを用いて実装した．サーバ側プロキシは Python で動作する CGI プログラムとして実装した．ステガノグラフィにより情報を埋め込み・抽出する処理には Stepic [12]を用いた．Stepic は PNG 画像の最下位ビットに情報を埋め込んでいる．Stepic は画像処理に Python Imaging Library を利用している．プロキシ間の通信には libcurl の Python ラッパである PyCurl [13]を使用した．Stepic はバージョン 0.3，Python Imaging Library はバージョン 1.1.7，PyCurl はバージョン 7.19 を用いた．クライアント側プロキシとサーバ側プロキシを動作させる OS として Linux2.6 を用いた．

プロトタイプ実装の動作確認を行った結果を示す．検証時のネットワーク構成を図 3 に示す．検証では図 3 の A の位置に監視システムがあると想定した．まず，テスト用の HTML ファイルを設置したウェブサーバを起動した．続いて，サーバ側プロキシ 1 を起動した．クライアント側プロキシを，サーバ側プロキシ 1 の IP アドレスとポート番号，CGI プログラムのパスを指定して起動した．ウェブブラウザの HTTP プロキシをクライアント側プロキシに設定し，ウェブサーバ実行マシンにある HTML ファイルを閲覧した．サーバ側プロキシとクライアント側プロキシの間のパケット（A の監視地点）をクライアント側プロキシ実行マシンのパケットキャプチャソフトウェアを用いて解析した．オリジナルのメッセージが平文で含まれず，ステガノグラフィによって画像に埋め込まれていることを確認した．

検証ではサーバ側プロキシ・クライアント側プロキシそれぞれにカバーデータとして PNG 画像を 3 個用意しランダムに選択して利用するようにした．サーバ側プロキシに用意したカバーデータのサイズはそれぞれ 80.5　KB，69.6KB，72.5KB であった．クライアント側プロキシに用意したカバーデータのサイズはそれぞれ 43.1KB, 17.8KB, 32.2KB であった．図 4 に HTTP のレスポンスを埋め込み前後のカバーデータの一例を示す．左の画像が元のカバーデータを示しており，右の画像は HTTP レスポンスを埋め込んだ後の画像の最下位ビットを示している．上部の帯状のノイズ部分に HTTP レスポンスのオブジェクト（HTML ファイル）が埋め込まれていることを確認した．

最後に開発したシステムを用いてローカルネットワークではない実際のインターネットのサイトを閲覧した．Google ニュースを閲覧した際の実行画面を図 5 に示す．HTML ファイルや小さなサムネイル画像は表示されているが YouTube の動画は ”An error occurred, please try again later”と表示された．このようにステガノグラフィの埋め込みができない大きさのファイルの場合にはプロキシが HTTP のエラーコード 403（Forbidden）を返すように実装できていることを確認した．

8. 性能評価

性能評価として Wikipedia の日本語版トップページ，およびそのトップページに埋め込まれているオブジェクトを取得したときの所要時間を計測した．Wikipedia 日本語版のトップページには PNG 画像ファイルが 17 個，JavaScript ファイルが 6 個，JPEG 画像ファイルが 5 個，スタイルシートファイルが 3 個含まれていた．これらのファイルの平均サイズは 14.7KB であった．これらのファイルをオリジナルサイトからダウンロードし図 3 のウェブサーバ実行マシンに配置して実験を行った．利用したハードウェアとソフトウェアの仕様を表 1 に示す．性能評価にはネットワーク性能測定ソフトである jmeter2.4[14]を用いた．jmeter の測定ではスレッド数を 5，Ramp-up 期間を 1 秒，ループ回数を 20 回として計測を行った．Ramp-up 期間とはスレッドの生成にかける時間のことである．図 6 に性能評価の結果を示す．所要時間は取得ファイルごとの平均値を示した．

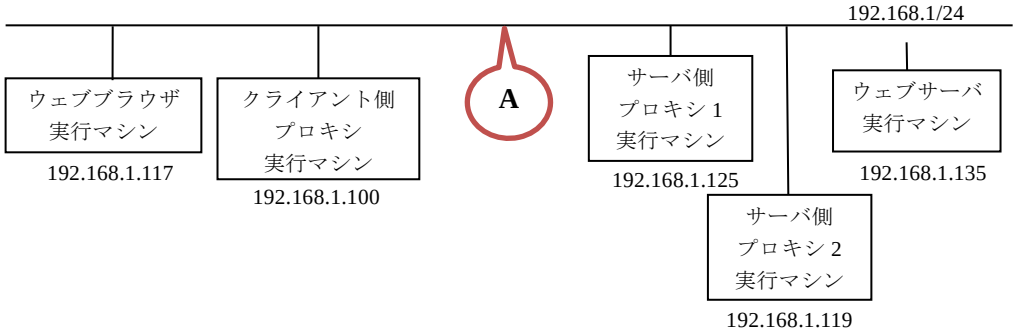

図 3 評価に用いたネットワーク構成

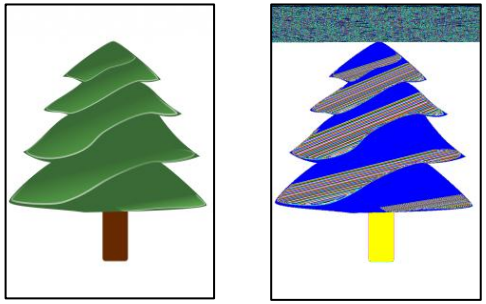

図 4　カバーデータ(左)と HTTP レスポンス埋め込み後の画像の最下位ビット部(右)

Vol.2011-DPS-146 No.11
Vol.2011-CSEC-52 No.11
2011/3/10

図 5　インターネットのサイトを閲覧した結果（Google ニュース）

表 1 性能評価に用いたハードウェアとソフトウェアの仕様

	Webブラウザ 実行環境	クライアント側 プロキシ 実行環境	サーバ側プロキシ1 実行環境	サーバ側プロキシ2 実行環境	ウェブサーバ 実行環境
OS	Windows 7 HomePremium 64bit	Linux 2.6.35	Linux 2.6.26	Linux 2.6.29	Linux 2.6.29
CPU	Intel(R) Core(TM)2 Duo CPU U9400 1.4GHz	Intel(R) Core(TM)2 CPU 6300 1.86GHz	Intel Pentium M processor 1.2GHz	Intel(R) Pentium(R) M processor 1500MHz	Intel(R) Celeron(R) M processor 1.40GHz
メモリ	4GB	2GB	1GB	1GB	512MB
NIC	Intel(R) 82567LM Gigabit Network Connection	Broadcom Corporation NetXtreme BCM5755 Gigabit Ethernet PCI Express	Intel PRO/100 VE Network Connection	Intel Corporation 82540EP Gigabit Ethernet Controller (Mobile)	Marvell Technology Group Ltd. 88E8036 PCI-E Fast Ethernet Controller
ソフトウェア	JMeter2.4	Python 2.6.6で実装した クライアント側プロキシ	Python 2.5.2で実装した サーバ側プロキシ	Python 2.5.2で実装した サーバ側プロキシ	Python 2.6.2付属の SimpleHTTPServer

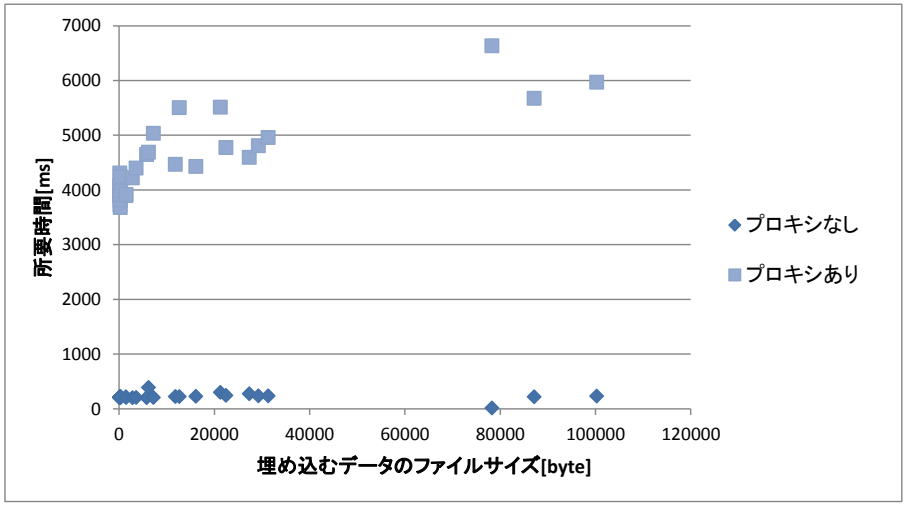

図 6 Wikipedia 日本語版トップページ取得にかかった時間

9. 考察

本節ではまず性能評価の結果について考察する．続いて提案方式の利便性と通信の本物らしさについて考察する．最後に提案方式の検出方法について議論する．

9.1 性能評価の考察

提案方式はオリジナルの通信データをそれよりも大きなカバーデータの画像ファイルに埋め込んで送受信することから通信量が増加する．性能評価は負荷のかかっていない実験ネットワークで行ったため，上記の通信時間の増加の影響はほとんど現れていない．つまり今回の性能評価でグラフに表れている処理時間の増加はクライアント側プロキシとサーバ側プロキシによる処理時間の増加分であると考えられる．埋め込まれているファイルの大きさによって所要時間に大きな差が見られないのは，増加分の処理時間の大半を画像ファイルの読み込みや Python インタプリタによるプログラムの起動などに費やしていることが示唆されている．実際の通信環境では通信量の増加によって所要時間がさらに増加すると考えられる．

処理性能を高めるためには，サーバ側プロキシを Python インタプリタで動作する CGI プログラムとしてではなく高速なサーバプログラムとして実装する必要がある．カバーデータの読み込み時間やクライアント側プロキシとサーバ側プロキシの間の通信時間を減らすために，送信する情報量に応じてサイズの違うカバーデータを可変で

使い分けることで通信量の増加をある程度抑制できると考えられる．

9.2 利便性

提案システムは２種類のプロキシソフトウェアとして実装した．HTTP プロキシとして実装することで，特別なウェブブラウザを使うことを利用者に強制せず済み，利用者はプロキシでの遅延時間を除けば，普段の使用感と変わらず，秘匿通信を行うことが出来るようになった．提案方式の欠点としては事前に監視拠点の外側にサーバ側プロキシを設置する必要があるということである．既に監視されている人物が，監視拠点の外側へサーバを設置することは困難である．当然外部のプロキシの存在を知られた時点でサーバ側プロキシに通信記録が記録されていれば，過去の通信の秘匿は無効になってしまう．またウェブブラウザとクライアント側プロキシの間，ウェブサーバとサーバ側プロキシの間は通信が秘匿されていないことからこれらの通信の監視には対応できない．

9.3 通信の本物らしさ

今回の実装ではプロキシ間の通信はクライアント側プロキシが画像を送りサーバ側プロキシが画像を送り返す通信パターンになっている．通常の通信では HTTP の POST メソッドで画像をウェブサーバに送信したとき，レスポンスは HTML 文書であることが普通である．今回は実装を簡単にするために画像を POST して，結果も画像が返ってくる通信パターンを採用した．画像を送受信する通信というのは簡単に検出できる明らかに不自然な通信パターンである．本稿で示したプロトタイプ実装を実用化するためには画像でリクエストを送信する部分はそのままで構わないが，レスポンスを取得する部分を改良する必要がある．具体的にはサーバ側プロキシの通信パターンを画像共有サイト等に見せかけて，HTTP レスポンスを一旦 HTML 文書の形で返し，HTML 文書内のリンクが指す画像データにレスポンスを埋め込むことによって，外部から見たときに画像を投稿し現在投稿している画像の一覧を受け取っているという風な，より自然な通信パターンに見せかける方法が考えられる．

9.4 提案方式の検出　～　監視システムの立場から

本稿ではここまで通信を秘匿する立場で議論をしてきたが，提案システムはその性質上悪用されることが十分に考えられる．そこで悪用を防ぐ立場として提案システムを検出する方法について考察を行う．提案方式の秘匿通信を検出するためにネットワーク上を流れるすべてのオブジェクトに対してステガノグラフィ情報が埋め込まれているかを検査することが考えられる．ステガノグラフィ技術は基本的にどのようなメディア（音声，画像，動画，テキスト）もカバーデータとして利用できる．よって流通する膨大な通信量のコンテンツ全てに対してリアルタイムでステガノグラフィの検査を行うことは現実的でない．我々はステガノグラフィを用いた秘匿通信の検出方法として以下の方法を提案したい．まず標準的な利用者の典型的なウェブサイトへの通信パターンを定義する．そして標準的な通信パターンから閾値を超えるほど例外的な通信を繰り返している場合にのみ，ステガノグラフィ技術を用いている可能性が高いと判断してデータを検査する．このように，提案方式の現実的な検出方法としては，まず通信パターンを監視し，パターンが異常である通信に対してだけステガノグラフィ情報が埋め込まれているかどうかを詳細に検査する段階的な方法が現実的であると考えられる．

10. おわりに

本研究では，インターネットにおけるウェブ通信の秘匿を実現する一方式としてステガノグラフィ技術を用いて通信を秘匿するプロキシの設計を示し，プロトタイプ実装の性能評価をおこなった．最後に提案方式の検出方法として通信パターンに着目して特定の通信のみデータ検査をおこなう方法について述べた．今後はステガノグラフィを用いた秘匿通信の研究を進め，並行して監視システムでの効率的な検出について検討していきたい．

参考文献

1) The Wall Street Journal, Iran's Web Spying Aided By Western Technology, June 22 (2009).
2) Thomas M. Chen and Victoria Wang, Web Filtering and Censoring, IEEE Computing now, March, pp.94-97 (2010).
3) Allot Communications, Digging Deeper Into Deep Packet Inspection (DPI), Allot Communications (2007).
4) Anonymizer, http://www.anonymizer.com/.（2011 年 2 月 14 日閲覧）
5) Global Internet Freedom Consortium, GTunnel, http://www.internetfreedom.org/.（2011 年 2 月 14 日閲覧）
6) JonDonym, http://anonymous-proxy-servers.net/.（2011 年 2 月 14 日閲覧）
7) David M. Goldschlag, Michael G. Reed, and Paul F. Syverson, Hiding Routing Information, Information Hiding, Springer-Verlag LLNCS 1174, pp.137-150 (1996).
8) Leavitt, N., Anonymization Technology Takes a High Profile, IEEE Computer, November, pp.15-18, (2009).
9) Tor project, http://www.torproject.org/.（2011 年 2 月 14 日閲覧）
10) Katzenbeisser, S., and Petitcolas, F., Information hiding techniques for steganography and digital water marking, ARTECH HOUSE (2000).
11) The Internet Society, RFC2616: Hypertext Transfer Protocol -- HTTP/1.1 (1999).
12) Domnitser, L., Stepic – Python image steganography, http://domnit.org/stepic/.（2011 年 2 月 14 日閲覧）
13) Jacobsen, K., and Oberhumer, M., PyCurl, http://pycurl.sourceforge.net/（2011 年 2 月 14 日閲覧）
14) Apache Jakarta™ Project, Apache JMeter, http://jakarta.apache.org/jmeter/（2011 年 2 月 14 日閲覧）