Vol.2009-IOT-6 No.3
2009/6/27

SCTPを用いたネットワークにおけるサイドチャネル解析対策の提案

三　村　　守[†1]

LANを構成するホストの脆弱性は，OS (Operating System) やアプリケーションに密接に関係している．LAN内部で稼動するOS，アプリケーションの種類やバージョンに関する情報が外部に漏洩すれば，その情報によってLAN内部で稼動するホストの脆弱性が悪意ある第三者に知られる可能性がある．既存の暗号通信では通信内容を秘匿することはできるが，OS，アプリケーションの種類等のサイドチャネル情報を秘匿することはあまり考慮されていない．本稿では，トラフィック分析技術を悪意ある第三者が利用した場合に何が脅威となるかについて述べ，ネットワークにおけるサイドチャネル解析への対策を実装手法も含めて検討する．そして，脆弱性を悪意ある第三者に知られないことを目的とするSCTPを用いたサイドチャネル解析対策を提案する．さらに，サイドチャネル解析対策を施したVPNアプリケーションを試作し，検証実験により性能を評価する．

Proposal of a Countermeasure using SCTP against Remote Side-channel Attacks

Mamoru Mimura[†1]

Vulnerabilities of a host that constructs a LAN relates the OS (Operating System) or the application. The OS or the application may leak out and yield vulnerabilities to a malicious third person. Cryptographic communication can conceal the contents. However, previous cryptographic communication does not take notice of concealing such as side-channel information. This report explains threats of traffic analysis and examines countermeasures that include how to implement against remote side-channel attacks. Then, we propose a countermeasure using SCTP against remote side-channel attacks in order to conceal vulnerabilities from a malicious third person. Moreover, we develop a VPN application against remote side-channel attacks and the verification experiments show the performance.

1. はじめに

ネットワークに接続するホストの脆弱性は，OS (Operating System) やアプリケーションに密接に関係している．OSやアプリケーションの脆弱性は毎日のように報告されており，修正プログラムの配布も追いつかない場合がある．また，まだ報告されていない脆弱性を利用した攻撃も多く発生するようになってきている．このように，OSやアプリケーションは常に修正すべき問題を抱えているものと考えられる．よって，LAN内部で稼動するOSやアプリケーションの種類やバージョンに関する情報が外部に漏洩すれば，その情報によってLAN内部で稼動するホストの脆弱性が外部に知られることになる．悪意ある第三者はトラフィック分析技術を駆使し，LAN内部で稼動するホストの脆弱性を知り，LAN内部への攻撃に役立てることができる．

本稿では，LAN外部に悪意ある第三者の存在を仮定し，LANのセキュリティを確保するために脆弱性を外部に知られないようにすることを研究の目的とする．以下，第2章ではトラフィック分析技術を悪意ある第三者が利用した場合に何が脅威となるかを述べ，挙動分析技術への対策の必要性について考察する．第3章では脆弱性を悪意ある第三者に知られないことを目的とするSCTP[13]を用いたサイドチャネル解析対策を提案する．第4章では検証実験により試作したアプリケーションの性能を評価し，最後にまとめと今後の課題を述べる．

2. トラフィック分析技術の脅威と対策

従来のトラフィック分析技術は，パケットに含まれるヘッダやペイロードに含まれる特定文字列をパターンマッチングにより検出し，プロトコルやアプリケーションを識別する手法が主流であった．しかし近年，トラフィック量や暗号通信の増加により，従来手法によるトラフィックの分類は困難になりつつある．そこで，ペイロードを走査せずに，パケット長や送受信タイミングなどに着目したトラフィック分析技術が研究されるようになった．トラフィック分析技術を，分析する対象を基に分類すると，ペイロード分析，ヘッダ分析および挙動分析の３つの手法に分類することができる[9]．本稿では挙動分析に着目し，悪意ある第三者が利用した場合の脅威と対策について考察する．また，暗号通信実装に対するサイド

†1 海上自衛隊
Japan Maritime Self-Defense Force

チャネル攻撃の可能性について示す.

2.1 挙動分析

暗号通信の普及に伴い，パケット長やパケット送受信のタイミング等のトラフィックの挙動を分析することによる，低負荷で高速なプロトコルやアプリケーションの推定技術が研究されるようになった．文献 1)，2) では単純に最初の数個のパケット長のみを利用し，高速で高精度なアプリケーション推定を実現している．文献 6)，12) では，パケット長やタイミング等のフロー挙動がアプリケーションの種類によって異なる事を利用したアプリケーション推定手法が提案されている．文献 7)，16) ではパケット長や方向等の属性値の遷移パターンを抽出し，トラフィックを推定する手法が提案されている．これらの挙動分析によるトラフィック分析手法では，ペイロードやヘッダを走査する必要がなく，パケット長およびその到着時間という限られた情報から，高精度でアプリケーションやプロトコルを推定することを実現している．このように，ペイロードやヘッダを分析せず，挙動を分析することにより LAN 内部で稼動するホストの OS やアプリケーション推定する技術は実用的となりつつある．したがって，挙動分析により流出する情報とそれにより考えられる脅威としては，ヘッダ分析の場合と同様に LAN 内部で稼動するホストの脆弱性を知られ，攻撃者による侵入を引き起こす可能性が考えられる．

2.2 暗号通信実装に対するサイドチャネル攻撃

従来の暗号に対する攻撃法では，既知平文攻撃や選択暗号文攻撃のように，平文や暗号文は入手できるが，処理中のデータは入手できないことが前提となっていた．しかしながら，攻撃者が暗号処理の時間や消費電力を精密に測定できる場合には，副次的なサイドチャネルから漏洩する情報も考慮することが必要である．文献 4) では OpenSSL を用いた Web サーバに対するタイミング攻撃が有効であり，悪意ある第三者がネットワークを経由して遠隔で秘密鍵を抽出する手法が示されており，対策の必要性が提唱されている．このように，タイミング攻撃に対する脆弱性は，暗号通信実装に致命的な影響を与える場合もある．遠隔によるタイミング攻撃が可能となる原因の 1 つはネットワークの実装にあり，暗号通信実装に対するサイドチャネル攻撃とも言うことができる．現在のネットワークアーキテクチャではトラフィックを盗聴し，挙動分析を実施することは容易であるため，悪意ある第三者がトラフィックパターンを分析することも考慮する必要がある．サイドチャネルにはトラフィックパターンのようなネットワークの挙動も考えられ，他の暗号通信実装に対するサイドチャネル攻撃の可能性も否定できない．

2.3 挙動分析対策

ペイロードやヘッダの走査を必要としないアプリケーションの推定技術は比較的新しく，アプリケーションの推定に対する直接の対策手法はあまり検討されていない．数少ないトラフィック分析への対策としては，パディングによりパケット長を変更し，トラフィックパターンを変化させる手法が提案されている[14),15)]．しかしながら，匿名通信の研究分野ではトラフィック分析への対策技術は古くから研究されている．文献 8) のようなデータリンクパディングと呼ばれるパケット長を変化させる手法は，多くの研究者によって提案されている．文献 3) ではダミートラフィックを挿入し，トラフィックパターンを秘匿する手法が提案されている．また，文献 11) では，ダミートラフィックを減らすために，パケット送信のタイミングを変更する手法も検討されている．既存のトラフィック分析への対策手法をまとめると，パディングや分割によるパケット長の変更，ダミートラフィックの挿入およびパケット送信のタイミングの変更に分類される．前章で考察した脅威を考慮すると，挙動分析に対してもヘッダ分析と同程度の対策が必要であると考えられる．しかしながら，匿名通信の研究分野におけるトラフィック分析対策は，主に誰もが参加できるパブリックなネットワークにおける匿名性の保護を目的としたものであり，特定の利用者のみが参加するプライベートなネットワークにおいて，悪意ある第三者に脆弱性に関する情報を知られないことを目的とする場合には最適とは限らない．また，どのようにパケット長やタイミングを変更するかという議論は活発であるが，どのようにしてパケット長やタイミングを制御するアプリケーションを実装するかについてはあまり議論されていない．そこで，匿名性の保護を目的とする対策手法を応用し，LAN 外部に脆弱性を知られないことを目的とする挙動分析への対策を実装手法も含めて検討する．

3. 脆弱性を知られないためのサイドチャネル解析対策

この章では，LAN 外部に脆弱性が知られる原因を考察し，脆弱性を知られないことを目的とする SCTP を用いたサイドチャネル解析対策を提案する．また，ネットワークにおけるサイドチャネル解析対策を施した VPN を試作する．

3.1 脆弱性が知られる原因

通信内容が暗号化され，暗号を解読することができないと仮定した場合，盗聴者が利用することができる情報はパケット長とその方向および到着時間に限られる．また，TCP/IP のプロトコルレイヤにおいては，適用した暗号通信プロトコルより上位層の情報が暗号化されるため，盗聴者は下位層の情報を利用することができる．パケット長とその方向および到

着時間から得られる情報は，適用する暗号通信の階層に関わらず利用可能である．VPN 等によりパケットがカプセル化され，パケット長が変化した場合にも，複数のパケット間の相対的なパケット長の差を分析することにより，パターンを検出することが可能であると考えられる[5]．すなわち，アプリケーションやプロトコルが推定され，LAN 外部に脆弱性が知られる原因の１つは，パケット長とパケット送受信のタイミングに特定のパターンを生じさせているネットワークの実装である．これは，暗号通信実装に対するサイドチャネル攻撃が可能となる原因と一致する．したがって，トラフィックパターンを秘匿することは，ネットワークにおけるサイドチャネル解析対策ともなり得る．ネットワークにおけるサイドチャネル解析対策を施すことで，LAN 外部に脆弱性が知られるのを防ぎ，深刻な脅威を回避することができる．

3.2 トラフィックパターンの変更

トラフィックパターンを変更するために，パケット長とタイミングを一定かランダムに変更することを検討する．パケット長に関しては，どちらの場合も実装における課題は少ない．一定の場合には効率がやや悪くなるが，スループットには致命的な影響はないものと考えられる．また，パケット長のパターンが完全になくなるため，ランダムにする方式よりも高い強度が期待できる．したがって，パケット長については一定にする方式とする．タイミングに関しては，スループットを考慮すると，一定にする方式の実装は困難であると考えられる．ある程度のスループットを確保するためには，一定である送信間隔は可能な限り短くする必要がある．パケットが継続して到着する場合，送信間隔は遅延時間に加算されるためである．しかしながら，実用的なスループットが確保できるように送信間隔を短くした場合，効率は著しく低下し，ネットワークへ与える負荷も増大する．これは，インターネットで運用することを考慮した場合には許容できない．したがって，タイミングは遅延を最小限にしつつランダムにする方式とする．

3.3 ネットワークにおけるサイドチャネル解析対策の提案

各々の端末において，パケット長およびパケット送受信のタイミングによるトラフィックパターンを生じさせないようにするためには，OS やアプリケーションのネットワークの実装を変更すればよい．しかしながら，これをネットワークに接続するあらゆるホストに適用するのは現実的ではない．また，挙動分析への対策が必要となる高い秘匿性を求められる機会は限られている．そこで，ネットワークの経路上においてサイドチャネル解析対策を施した VPN を検討する．一般的にインターネットでの利用を前提とする VPN アプリケーションは信頼性を確保するために TCP を用いて実装される．しかしながら，トラフィックパターンを制御することを考慮した場合，TCP はストリーム指向であるため，アプリケーションにおいてパケット長を制御する実装は困難である[10]．データグラム指向である UDP を利用すればパケット長は容易に制御できるが，インターネットで VPN を構築するためには信頼性が求められる．そこで，アプリケーションにおいてパケット長の制御が可能であり，信頼性を確保することができる SCTP を用いたネットワークにおけるサイドチャネル解析対策を提案する．SCTP は多くの UNIX 系の OS で導入されており，Windows においてもライブラリ等を利用することで動作する．以下，その手法について**図 1** を用いて説明する．送信元ゲートウェイは，パケットを一定の長さに分割して暗号化し，送信バッファに格納する．分割したデータが固定長に満たない場合にはパディングを実施する．送信バッファに格納されたデータをランダムな遅延を付加して宛先ゲートウェイへ送信する．宛先ゲートウェイはカプセル化されたパケットを受信し，元のデータに復号化して受信バッファに格納する．受信バッファに元のパケットを構成するすべてのデータが集まると，元のパケットを復元して本来の宛先に中継する．これにより，送信側ゲートウェイと宛先ゲートウェイの間でトラフィックパターンを変えることができる．この動作を相互に実施することで，アプリケーションやプロトコルが推定される原因となる本来のトラフィックパターンを秘匿し，ネットワークにおけるサイドチャネル解析対策を施した VPN を実現することができるものと考える．

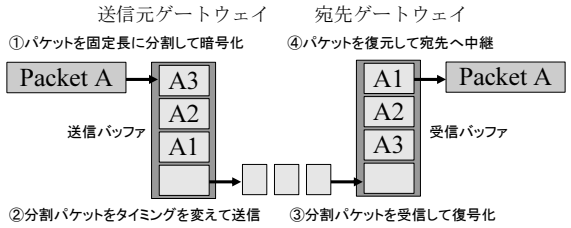

図 1 トラフィックパターンを変えるゲートウェイ
Fig. 1 Gateways that change traffic patterns

3.4 サイドチャネル解析対策を施した VPN の試作

ネットワークにおけるサイドチャネル解析対策を施した VPN を，C 言語を用いて Linux (Fedora 10) が動作するシステムにおいて試作した．試作したプログラムの主な開発環境お

よび仕様を**表 1** に示す．通信内容の暗号化には AES を利用し，Mersenne Twister で生成した乱数およびハッシュ関数によりランダムな遅延を付加する．よってパケット長は AES のブロックサイズの整数倍の一定の長さとなる．送信間隔はスループットへの影響を考慮し，バッファに 50%以上の空き容量がある場合には 0〜10ms の遅延を付加する．プロトコルは UDP および SCTP から選択可能とし，元のパケットの IP ヘッダ以降を分割の対象とする．SCTP を選択した場合のゲートウェイ間のパケットは**図 2** に示すようにカプセル化される．

表 1 開発環境および仕様
Table 1 Development environment and specifications

OS	Linux (Fedora 10)
プログラム言語	C 言語 (gcc-4.3.2)
暗号	AES
乱数	Mersenne Twister
パケット長	128/256/512/1024 bytes
送信間隔	0〜10 ms
トンネリング層	L3 (Network 層)
プロトコル	UDP/SCTP

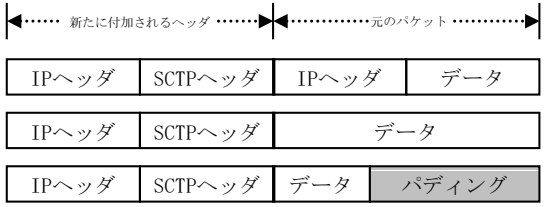

図 2 カプセル化されたゲートウェイ間のパケット
Fig. 2 Encapsulation for packets between gateways

4. 検証実験

この章では，実験ネットワークにて検証実験を実施し，試作したネットワークにおけるサイドチャネル解析対策を施した VPN の性能を評価する．

4.1 実験環境

図 3 に示す実験ネットワークを利用して試作したプログラムを検証する．図中の各ホスト間は 100BASE/T イーサネットで接続されており，図中に示す OS が動作している．ルータである GW1 および GW2 で試作したプログラムを動作させ，この間を FreeBSD に標準で組み込まれている dummynet を介して VPN で接続する．dummynet は帯域，遅延時間，パケット損失率等の制御を行うことが可能であり，これを利用してインターネットの環境を模擬する．SV では電子メール，DNS，Web，時刻同期等のサービスを提供するアプリケーションを動作させ，host1〜3 からこれらのサービスを利用させる．

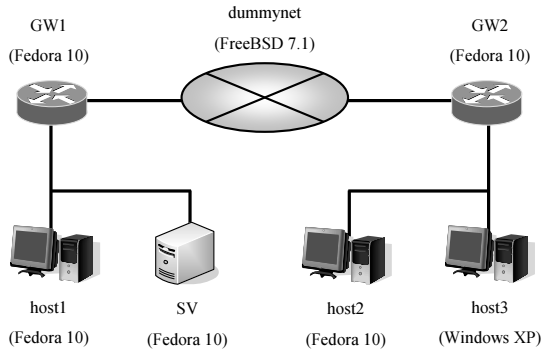

図 3 実験ネットワーク
Fig. 3 An experimental network

4.2 実験内容

4.2.1 機能確認

SV で提供する電子メール，DNS，Web，時刻同期等のサービスを，host1〜3 から問題なく利用できることを確認する．

4.2.2 通信遅延

ping コマンドを利用し，SCTP を用いた場合の host2 と SV 間の RTT (Round Trip Time) を計測する．プロトコルを UDP，パケット長を 1024byte とした場合の RTT を基準値である 100 とし，100 回の平均値を相対値で算出する．この際に dummynet を利用して遅延時間およびパケット損失率を変化させる．

4.2.3 処理能力

ネットワークのトラフィックを測定するソフトウェアである Iperf2.0.4 を利用し，SCTP を用いた場合の host2 と SV 間のスループットを計測する．RTT の計測と同様に，プロトコルを UDP，パケット長を 1024byte とした場合のスループットを基準値である 100 として相対値を算出する．この際に dummynet を利用して遅延時間およびパケット損失率を変化させる．

4.3 実験結果

4.3.1 機能確認

表 2 にプロトコルごとの機能確認の結果を示す．電子メール，DNS，Web，時刻同期等のプロトコルが，試作したトラフィックパターンを変えるゲートウェイを介して問題なく動作することを確認した．

表 2 機能確認の結果
Table 2 Results of a functional test

Protocol	確認結果
ICMP	○
SMTP, POP, IMAP	○
DNS	○
HTTP, HTTPS	○
NTP	○

4.3.2 通信遅延

SCTP における各パケット長ごとの RTT の相対値を**図 4** に示す．図の縦軸は RTT の相対値，横軸はパケット長を表す．パケットの損失がない場合，RTT はパケット長に関わらずほぼ一定となった．遅延時間が 0 の場合の RTT は基準値である 100 に近く，SCTP を用いることによる通信遅延に対するオーバヘッドはほとんどないことが確認できた．5 %の損失率を与えた場合には，RTT は基準値の 10 倍以上の大きな値となった．これは，パケットの損失に伴い，再送要求が発生したためであると考えられる．

4.3.3 処理能力

SCTP における各パケット長ごとのスループットの相対値を**図 5** に示す．図の縦軸はスループットの相対値，横軸はパケット長を表す．パケットの損失がなく，パケット長を 256〜1024byte に設定した場合のスループットは基準値に近いほぼ一定の値となった．パケット長を 128byte に設定した場合には，スループットは基準値の約半分以下の値となった．この結果から，パケット長を 256byte 以上にすれば，ある程度のスループットが期待できることが確認できた．5 %の損失率を与えた場合にはパケット長に関わらずスループットは不安定となり，基準値の半分以下の低い値となった．

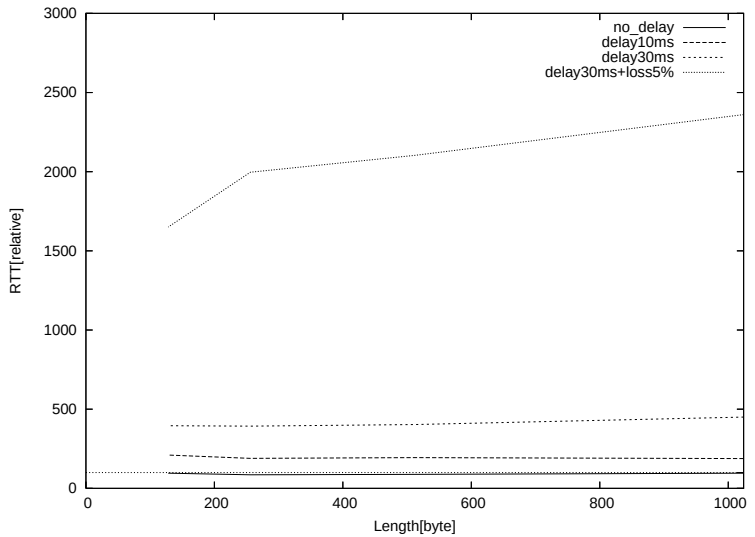

図 4 RTT の相対値
Fig. 4 Relative values of RTT

5. おわりに

本稿では，トラフィック分析技術を悪意ある第三者が利用した場合にどのような情報が漏洩し，何が脅威となるかを明らかにし，ネットワークにおけるサイドチャネル解析への対策を実装手法も含めて検討した．さらに，脆弱性を LAN 外部に知られないことを目的とする SCTP を用いたサイドチャネル解析対策を提案し，検証実験により試作した VPN アプリケーションの性能を評価した．

試作したネットワークにおけるサイドチャネル解析対策を施した VPN では，SCTP を用いることでインターネットでの運用を可能とする信頼性を確保し，パケット長とタイミング

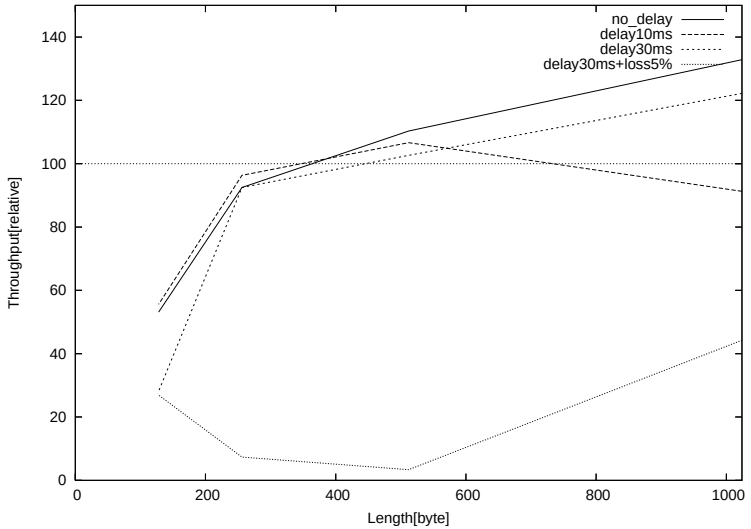

図 5 スループットの相対値
Fig. 5 Relative values of throughput

を制御することを実現した．本稿では，どのようにしてパケット長とタイミングを制御するのかを示したが，どのようにパケット長とタイミングを変更するかについての検討は十分ではない．パケット長を固定長に変更する方式は，あまり効率が良いとは言えない．パケット長を固定長に変更したとしても，送受信量の関係からアプリケーションを推定されてしまう可能性がある．送受信量の関係を秘匿するためには，ダミートラフィックを挿入することも検討する必要があるものと考える．通信効率とサイドチャネル解析対策の効果はトレードオフの関係にあり，いかに効率的にサイドチャネル解析対策を実装するかは大きな課題である．

参 考 文 献

1) Bernaille, L., Teixeira, R., Akodkenou, I., Soule, A. and Salamatian, K.: Traffic Classification On The Fly, *ACM SIGCOMM Computer Communication Review*, Vol.36, pp.23–26 (2005).
2) Bernaille, L., Teixeira, R. and Salamatian, K.: Early Application Identification, *In the 2006 ACM CoNEXT Conference*, No.6 (2006).
3) Berthold, O. and Langos, H.: Dummy Traffic Against Long-term Intersection Attacks, *Proceedings of the 2nd International Workshop on Privacy Enhancing Technologies*, Vol.2482, pp.110–128 (2002).
4) Brumley, D. and Boneh, D.: Remote Timing Attacks Are Practical, *Proceedings of the 12th Conference on USENIX Security Symposium*, pp.1–14 (2003).
5) Gebsk, M., Penev, A. and Wong, R.K.: Protocol Identification of Encrypted Network Traffic, *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pp.49–54 (2006).
6) 北村　強，静野隆之，岡部稔哉：フロー挙動分析技術に基づくアプリケーション識別手法，信学技報 NS2005-136， Vol.105, No.470, pp.13–16 (2005).
7) 北村　強，静野隆之，岡部稔哉：パケットタイプ遷移パタン分析を用いたトラヒック識別手法，信学技報 NS2006-27， Vol.106, No.41, pp.25–28 (2006).
8) Liberatore, M. and Levine, B.N.: Inferring the Source of Encrypted HTTP Connections, *Proceedings of the 13th ACM Conference on Computer and Communications Security*, pp.255–263 (2006).
9) 三村　守：脆弱性を知られないためのネットワークにおけるサイドチャネル解析対策，2009 年暗号と情報セキュリティシンポジウム予稿集 3E3-3 (2009).
10) 三村　守，中村康弘：トラフィックフロー分析に耐性があるトンネリング手法の分析，情処研報 2007-CSEC-39， Vol.2007, No.126, pp.7–12 (2007).
11) Rennhard, M., Rafaeli, S., Math, L., Plattner, B. and Hutchison, D.: Analysis of an Anonymity Network for Web Browsing, *Proceedings of 11th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pp.49–54 (2002).
12) 静野隆之，北村　強，岡部稔哉：フロー挙動分析によるアグリゲーションフローのアプリケーション識別手法，信学技報 NS2005-160， Vol.105, No.627, pp.9–12 (2005).
13) Stewart, R.: RFC 4960 Stream Control Transmission Protocol (2007).
14) Timmerman, B.: A Security Model for Dynamic Adaptive Traffic Masking, *In the New Security Paradigms Workshop* (1997).
15) Timmerman, B.: Secure Dynamic Adaptive Traffic Masking, *Proceedings of the 1999 Workshop on New Security Paradigms*, pp.13–24 (1999).
16) 八木清之介，和泉勇治，角田　裕，根元義章：ペイロード長の遷移パタンを用いたネットワークアプリケーション弁別手法，情処研報 2007-DSM-45， Vol.2007, No.38, pp. 83–88 (2007).