情報処理学会論文誌　Vol. 52　No. 9　2853–2865 (Sep. 2011)

トラフィックパターンを隠すアプリケーションベースVPNの実現方式

三村　守[†1,†2]　田中英彦[†1]

通信内容を調査する必要がないトラフィック分析技術は，暗号通信の分析に適用することが可能である．これらのトラフィック分析技術は暗号通信の内容を解読するものではないが，アプリケーションの種類等の副次的な情報を得ることを可能とする．よって攻撃者のトラフィック分析により，情報システムのアプリケーション等の脆弱性が外部に知られ，不正アクセスを引き起こす可能性が考えられる．しかしながら，OpenSSH，OpenVPN 等の既存のアプリケーションベース VPN では，通信内容を秘匿することはできるが，トラフィック分析から得られる副次的な情報を秘匿することはできない．本論文では，パケット長と送信間隔を変更することにより副次的な情報を秘匿し，トラフィックパターンを隠すアプリケーションベース VPN を試作し，実験によりその特徴を分析するとともに，実装における課題を明らかにする．さらに，実装における課題を解決するために，SCTP（Stream Control Transmission Protocol）を採用し，実装した VPN の性能を検証実験により評価する．

Implementation of an Application Based VPN that Conceals Traffic Patterns

Mamoru Mimura[†1,†2] and Hidehiko Tanaka[†1]

Traffic analysis technologies that do not scan the payload of communications can analyze encrypted traffic. Though these traffic analysis technologies do not decrypt the payload, enable to obtain secondary information e.g., application name. Thus, vulnerabilities of the application installed in the information systems are known outside by the attacker's traffic analysis, and it may cause unauthorized computer access. Though the previous application based VPN such as OpenSSH or OpenVPN protects the payload of communications, can not conceal the secondary information by the traffic analysis technologies. In this paper, we implement the application based VPN that conceals traffic patterns by altering packet sizes and the timing. Our experiments analyze the feature and reveal the problem to implement the application based VPN. In addition, to solve the problem, we adopt SCTP (Stream Control Transmission Protocol), and our verification experiments evaluate the performance of the VPN.

1. はじめに

近年，多くの情報システムが構築され，いくつかの分散システムはインターネットを通じて VPN（Virtual Private Network）で接続され，さらに複雑な情報システムを構成している．インターネットのような信頼性の低いネットワークを通じて情報システムを構成するために，VPN の役割は重要である．一般に，VPN の安全性は適用する暗号の強度に依存する．これらの情報システムは多くの利用者に利益をもたらすとともに，セキュリティリスクももたらす．多くの情報システムは脆弱性のような安全性のリスクをかかえている．今日，脆弱性は毎日のように報告されており，誰でもインターネットを利用して容易にそれらを見つけることができる．脆弱性は，情報システムで使用する OS（Operating System）やアプリケーションに密接に関係している．攻撃者は，情報システムの OS やアプリケーションを識別することができれば，容易に脆弱性を探すことができる．このような脆弱性を利用し，任意のコマンドを実行するためにはある程度の専門知識が必要である．しかしながら，専門知識がない者でもインターネットで公開されている実証コードを利用することにより，容易に脆弱性を利用することができる．また，Metasploit Framework[4] のような脆弱性検査のためのツールを悪用することも可能である．さらに，ゼロデイ攻撃のような未知の脆弱性を利用した攻撃も多発している．よって，情報システムの安全性を保証するためには，OS やアプリケーションを攻撃者に秘匿する必要がある．

他方，トラフィック量や暗号通信の増加から，パケット長や送受信タイミング等に着目した通信内容を調査する必要がないトラフィック分析技術が研究されるようになった．これらの分析技術は，トラフィックを把握する必要がある情報システムの管理者にとっては有益である反面，攻撃者が利用した場合には脅威となりうる．OpenSSH，OpenVPN 等の既存のアプリケーションベース VPN では，盗聴者に対して通信内容を保護することができるが，トラフィックパターンのような副次的な情報を保護することは考慮されていない．

†1 情報セキュリティ大学院大学
Institute of Information Security
†2 海上自衛隊
Japanese Maritime Self-Defense Force

本論文では，このようなトラフィック分析の脅威を明確にし，情報システムの安全性を保証するために，よりセキュアなアプリケーションベース VPN を開発することを目標とする．我々の目標はパブリックなネットワークにおける匿名性ではなく，プライベートなネットワークにおける秘匿性である．匿名通信の研究分野では，様々なトラフィック分析技術とその対策技術が研究されており，主な関心はどのようにトラフィックパターンを変えるかということである．しかしながら，どのようにその手法を実装するかについてはあまり議論されていない．本研究では匿名通信の研究分野におけるトラフィックパターンを変える手法を応用し，VPN の秘匿性向上のためにどのように実装するかについて検討する．

2. トラフィック分析の脅威

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推定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，ペイロードを調査せずに，パケット長や送受信タイミング等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．文献 43) では，通信を構成するパケットの長さ，方向および時間を可視化することにより，アプリケーションやプロトコルの特徴がパターンとして現れることを利用して，プロトコルの推定を行う手法が提案されている．文献 43) では他にも様々な可視化手法が提案されているが，いずれもパケット長およびその到着時間から得られる情報を利用していることに特徴がある．このようなトラフィック分析技術ではペイロードを調査する必要がないため，暗号通信の分析にも適用することが可能であると考えられる[12),16),45)]．文献 30) では平文と暗号化トラフィックにおけるパケット長，パケットの到着間隔等の統計情報の差分を分析しており，暗号化によって到着間隔を除く大部分の統計情報には大きな変化がないことが示されている．

このような挙動分析技術は，アプリケーションの推定にも応用されている．文献 5) 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 22) では，パケット長やパケット到着間隔等のフロー挙動の分析に基づくアプリケーション推定手法が提案されている．このアプリケーション推定手法では，**表 1** に示すフロー挙動パラメータが利用されている．フロー挙動パラメータは，パケット長およびその到着時間から得られる情報を利用している．文献 23) ではさらに，それらの情報をサポートベクトルマシンに機械学習させることにより，アプリケーションやプロトコルを推定する手法が提案されている．機械学習のトラフィック分類への応用については，ベイズ理論を用いた文献 32) 等他にも多くの試みがある[34)]．文献 21)，41) 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．これらの挙動分析技術では，ペイロードやヘッダを調査する必要がなく，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，トラフィックパターンを分析することにより情報システムの OS やアプリケーションを推定する技術は実用的となりつつある．したがって，挙動分析により情報システムの脆弱性を知られ，攻撃者による侵入を引き起こす可能性は否定できない．

表 1　フロー挙動パラメータ
Table 1　Parameters of flow behaviors.

パケット長		概要
パケット長	度数分布	パケット長の度数分布
	平均値	パケット長の平均値
	中央値	パケット長の中央値
	分散	パケット長の分布範囲
パケット到着間隔	度数分布	パケット到着間隔の分布範囲

3. 関 連 研 究

匿名通信の研究分野では，古くから挙動分析技術とその対策に関する研究が行われている．文献 9) ではメッセージを固定長とし，順序を入れ替える MIX と呼ばれるコンピュータ proxy が提案された．固定長のメッセージをつねに一定の間隔で送信することができれば，トラフィックパターンをなくすことができるものと考えられる．しかしながら，MIX ではつねに一定の間隔で送信するためのメッセージが得られるわけではない．ゆえに，意味のない偽のメッセージを送信してトラフィックパターンを隠す手法が提案された．多くの研究者がメッセージの大きさを秘匿するために，パディングによりトラフィックパターンを変化させる手法を提案している．文献 40) では，パディングによるオーバヘッドを減らしつつトラフィックを秘匿するモデルが提案されている．パディングの手法は，パケット自体の長さを変更する手法と，まったく別の偽のパケットを挿入する手法に分類される．文献 26) ではパディングによりパケットの長さを変更し，トラフィックパターンを変化させる手法が利用されている．文献 7)，14)，44) ではダミーパケットを挿入し，トラフィックパターン

を秘匿する手法が利用されている．匿名通信システムを通信速度を基に分類すると，High Latency（Message Based）システムおよび Low Latency（Connection Based）システムに分類される．High Latency システムは相互に作用しない電子メールのような通信を扱うものであり，Babel[17]，Mix Master[31]，Mix Minion[10] 等がある．Low Latency システムは相互に作用する Web，TELNET，インスタントメッセージのような通信を扱うものであり，Web MIXes[6]，Onion Routing[35]，Tor[11]，Crowds[36] 等がある．本論文で試作するトラフィックパターンを隠すアプリケーションベース VPN は，匿名性を目的とするものではないが，通信速度による分類では Low Latency システムに分類される．さらに，多くの研究者がタイミング分析とその対策技術について検討している[15),25),37),46)]．既存の挙動分析への対策をまとめると，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．

これらの対策技術は匿名性を目的としたものであり，攻撃者に情報システムの脆弱性を知られないことを目的としたものではない．匿名通信の研究分野においては，ダミーパケットは主に送信元と宛先の相関関係を隠すために利用され，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る．よってこれらの対策技術は，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，外部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．

秘匿性を目的とする VPN としては，IPsec[19] において挙動分析対策を施した TFC（Traffic Flow Confidentiality）パディングが提案されている[20)]．しかしながら TFC パディングでは，パケット長を減らす場合に，IPv6 の分割機能をそのまま活用している．そのため，攻撃者はパケットを再構築し，元のトラフィックパターンの推定を試みることが可能である．また，IPsec のようなネットワーク層の VPN は，それ自身でデータの到達を保証する機能を持たない．よって，インターネット等の信頼性の低いネットワークを通じて安全な通信経路を構築するためには，データの到達を保証する機能を持つ OpenSSH，OpenVPN 等のアプリケーションベース VPN も必要である．しかしながら，既存の OpenSSH，OpenVPN 等のアプリケーションベース VPN には挙動分析対策は施されていない．挙動分析対策を施したアプリケーションベース VPN は，文献 28) で提案されている．文献 28) では信頼性の低いネットワークを通じて安全な通信経路を構築するために，TCP を用いてアプリケーションベース VPN を試作しているが，TCP はストリーム指向のプロトコルであるため，アプリケーションにおいて任意にパケット長を制御することが困難である．

4. アプリケーションベース VPN の試作

この章では，文献 28) で提案されたアプリケーションベース VPN について説明し，実験によりその特徴を分析するとともに，実装における課題を明らかにする．

4.1 動作概要

既存のアプリケーションベース VPN[28] の動作概要を**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設定時間が経過するごとに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する．

4.2 試　　作

設計したトラフィックパターンを隠すアプリケーションベース 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．**図 2** を用いてその実装について説明する．試作したゲートウェイはクライアントおよびサーバから構成され，先に起動した側がサーバとなる．各々のゲートウェイは，外部と内部に 2 つのインタフェースを備えている．外部のインタフェースからはデータグラムソケットまたはストリームソケットを利用し，内部のインタフェースからはデータリンク層へのアクセスを提供する Linux 固有のパ

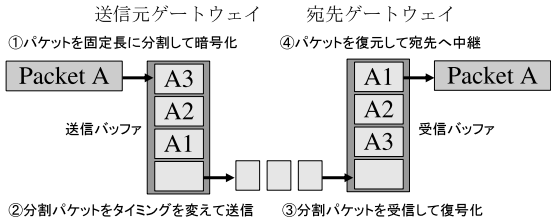

図 1　トラフィックパターンを隠すアプリケーションベース VPN
Fig. 1　An application based VPN that conceals traffic patterns.

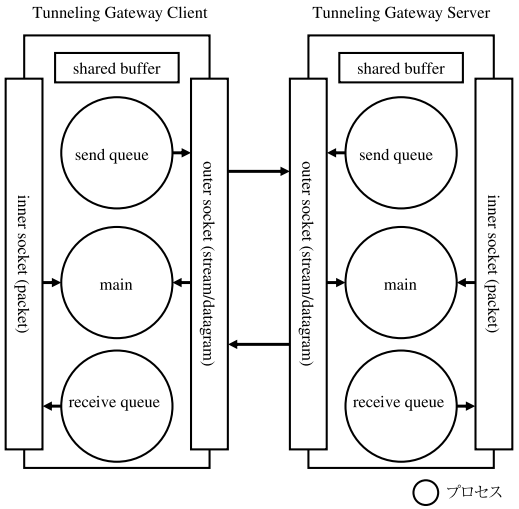

図 2　トラフィックパターンを隠すアプリケーションベース VPN の実装
Fig. 2　The implementation of the application based VPN that cpnceals traffic patterns.

ケットソケットを利用してパケットの送受信を行う．他の OS では，内部のインタフェースに各々のデータリンク層へのアクセスを提供するソケットを用いることで動作可能である．クライアントおよびサーバ間のプロトコルは UDP または TCP を使用し，内部のインタフェースから読み込んだパケットをカプセル化してサーバとクライアント間で送受信する．クライアントとサーバは同一の構造であり，それぞれ main，send queue，receive queue の 3 つのプロセスが内部で動作する．main プロセスは，select 関数により内部および外部のソケットを監視し，受信したデータを共有メモリに格納する．この際に，内部のソケットから受信したデータは，あらかじめ設定したパケット長に分割される．send queue プロセスは共有メモリを監視し，あらかじめ設定した送信間隔が経過するごとに，送信すべきデータがあれば外部のソケットに書き込む．receive queue プロセスは共有メモリを監視し，受信した先頭ペイロードに含まれる IP ヘッダからパケットを復元し，内部のソケットに書き込む．共有メモリにおけるキューの実装には，処理速度を考慮してリングバッファを採用した．

4.3 実　　験

試作したトラフィックパターンを隠すアプリケーションベース VPN を用いて実験を行い，その特徴を分析する．実験ネットワークの構成を**図 3** に示す．各ホストおよびルータは

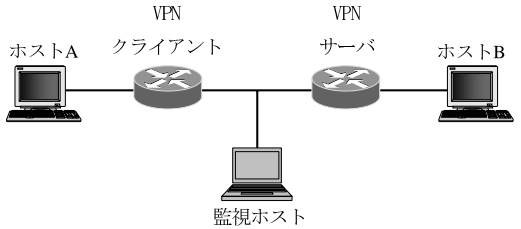

図 3　実験ネットワークの構成
Fig. 3　The network topology for the experiments.

Linux-2.6 システムを搭載した PC で構成されており，100BASE/T イーサネットで接続されている．各インタフェースにおける MTU（Maximum Transmission Unit）はいずれも 1,500 である．2 台のルータのカーネルのパケット転送機能は無効に設定されており，試作したアプリケーションベース VPN を動作させることによってホスト A とホスト B の間で VPN が構成され，通信が可能となる．ここで，クライアントおよびサーバ間のプロトコルに UDP または TCP を使用して VPN を構成し，ホスト A からホスト B に scp（Secure CoPy）コマンドで 1 MB のファイルを転送する．

4.3.1　パケット数の観測

まず，送信間隔を 0 に固定し，パケット長を変更することでパケット数がどのように変化するのかを測定した．パケット長を短く設定するほどパケットの分割数は多くなり，パケット数は増加するものと考えられる．**図 4** および**図 5** に設定パケット長ごとのパケット数の変化を示す．図の横軸は設定したパケット長であり，縦軸は観測したパケット数を示す．UDP の場合のパケット数に関しては，パケット長が短くなるほど増加しており，設計したとおりにパケットの分割が行われているものと考えられる．しかしながら，TCP の場合のパケット数に関しては，パケット長の変化に対してほぼ一定の値となった．パケットがあらかじめ設定したパケット長に分割されているのであれば，観測されるパケット数はパケット長が短くなるほど増加するはずである．この結果から，TCP の場合に関しては，意図したとおりにパケットの分割が行われていないことが確認できる．この原因は，実装では Nagle アルゴリズム[33] を無効にしているにもかかわらず，ソケットに書き込まれたデータがカーネルのスタックに蓄積され，ただちに送信されていないものと考えられる．

4.3.2　到着間隔の観測

次に，パケットの送信間隔を変更し，パケットの到着間隔を観測する．送信間隔を 0 に設

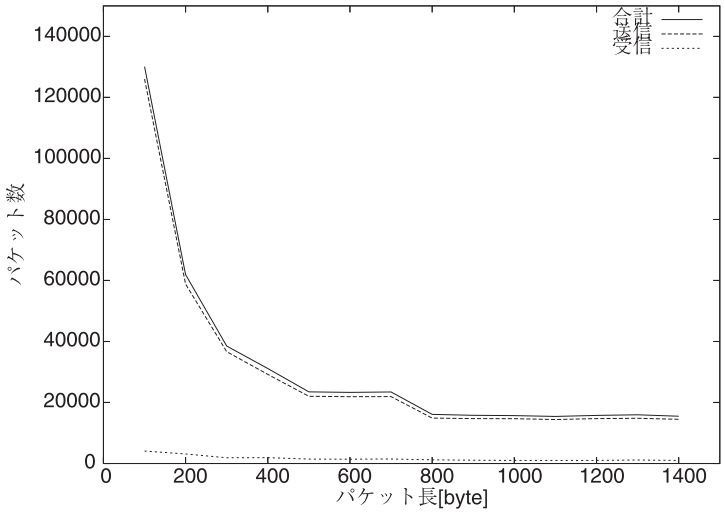

図 4　パケット長ごとのパケット数（UDP）
Fig. 4　A packet sizes-number graph (UDP).

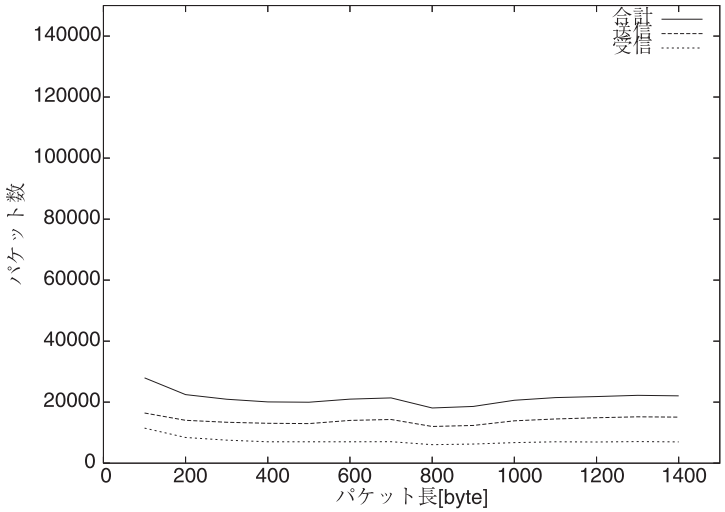

図 5　パケット長ごとのパケット数（TCP）
Fig. 5　A packet sizes-number graph (TCP).

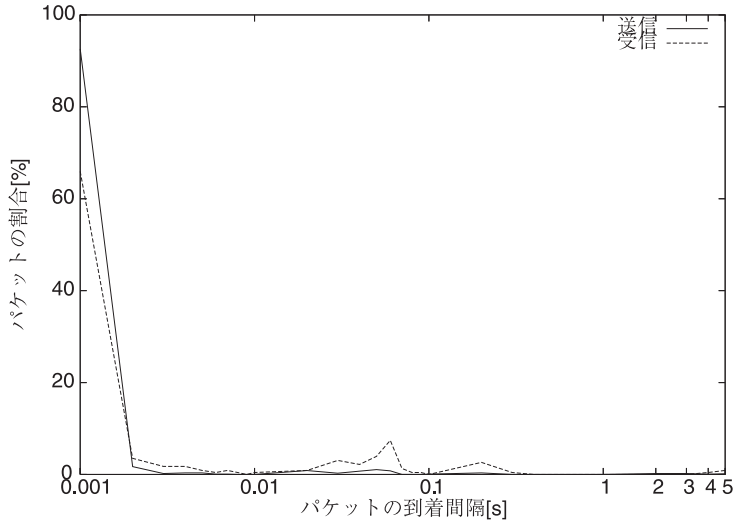

図 6　送信間隔 0 におけるパケット到着間隔の割合
Fig. 6　Rate of packet inter-arrival times (no delay).

定した場合のパケット到着間隔の割合を**図 6** に示す．図中の横軸は前のパケットとの到着時間の差であり，縦軸はその到着間隔に分類されるパケット数の全パケット数に対する割合である．0〜1 ms の送信間隔のパケットの割合が最も多く，送信は 90%以上，受信は 60%以上を占めている．このような数ミリ秒単位の送信間隔は，パケットの連続性と関係しているものと考えられる．すなわち，送信間隔が数ミリ秒単位の最大長のパケットの連続は，1 つの大きなデータを送信しているものと推定できる．送信間隔を 10 ms に設定した場合のパケット到着間隔の割合を**図 7** に示す．送信については 10〜20 ms の到着間隔のパケットの割合が最も多くなり，60%以上に変化した．これはパケットの送信間隔を 10 ms に設定したためである．しかしながら，受信については 30〜60 ms の到着間隔のパケットの割合がやや増加した．これは，送信側で遅延が発生することに起因し，遅れて到着する応答確認パケットの割合が増加したためと考えられる．この遅れて到着する応答確認パケットの割合は，送信間隔を長くするほど増加し，50 ms の送信間隔では 72.3%まで増加した．実験の結果，送信間隔を 10 ms に設定することで，送信間隔が数ミリ秒単位の短期間の挙動に対しては一定の効果があることを確認することができた．しかしながら，数ミリ秒以上の長期間の挙動

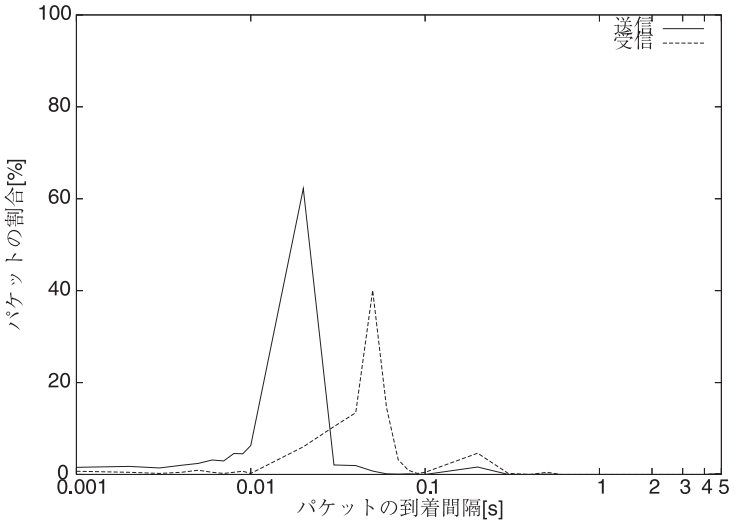

図 7　送信間隔 10 ms におけるパケット到着間隔の割合
Fig. 7　Rate of packet inter-arrival times (10 ms delay).

に関しては，制御することが困難であることを確認した．

4.4　実装における課題

4.4.1　パケット長の制御

UDP におけるパケット長に関しては，均一化されていることを確認できた．アプリケーションベース VPN の実装に，UDP 等のデータグラム指向のプロトコルを使用する場合には，パケット長を均一に制御することができる．しかしながら，UDP では通信の信頼性は保証されないため，パケットの損失や順序が入れ替わった場合の処置は別に実装する必要がある．TCP 等のストリーム指向のプロトコルを使用する場合には，通信の信頼性は保証されるが，Nagle アルゴリズムを無効にしても，パケット長を制御することは困難であった．TCP においてパケット長を制御するためには，ソケットにデータを書き込んだ後，Linux の実装では約 50 ms ほど待つ必要があった[28]．しかしながら，送信間隔を長くするほど，スループットは急速に低下する．そのため，TCP ではパケット長を制御しつつ，スループットを維持することは困難である．

4.4.2　送信間隔の制御

パケットの送信間隔に関しては，均一に制御することで観測されるパケットの到着間隔を変化させる効果があることを確認することができた．しかしながらその副作用として，遅れて到着する応答確認パケットの割合が増加し，Low Latency システムでは完全な制御は困難であることも確認した．よって Low Latency システムでは，送信間隔を均一に制御することで秘匿性を向上させることができるとはいい難い．それにもかかわらず，送信間隔を均一に制御することで，パフォーマンスには大きな影響が発生してしまう．また，意図的に送信間隔を制御しなくても，暗号化によるパケットの到着間隔の変化は比較的大きいという報告もある[30]．したがって Low Latency システムでは，送信間隔は均一に制御せずに，遅延を最小限にしつつランダムにする方式が最適であると考えられる．

5.　アプリケーションベース VPN の実装

5.1　SCTP によるアプリケーションベース VPN の実現方式の提案

一般的にインターネットでの利用を前提とするアプリケーションベース VPN 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．また，カプセル化する前の元のトラフィックが TCP の場合，元の階層および VPN の階層で 2 重に再送やフロー制御の機能が働き，パケット損失時に非効率となる可能性があること（TCP over TCP の問題）も指摘されている[18]．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，インターネット等の安全ではないネットワークにおいて VPN を構築するためには，信頼性が求められる．しかしながら，他のトランスポート層のプロトコルである RSVP[8]，DCCP[24] 等では信頼性を確保することは困難である．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（Stream Control Transmission Protocol）[38] を用いたアプリケーションベース VPN の実現方式を提案する．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．

5.2　実　　装

試作したアプリケーションを基に暗号機能を付加し，トラフィックパターンを隠すアプリケーションベース VPN を実装した[29]．実装したプログラムの主な開発環境および仕様を**表 2** に示す．通信内容の暗号化には AES[13] を採用し，Mersenne Twister[27] で生成した

表 2　開発環境および仕様
Table 2　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	128/256/512/1,024 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

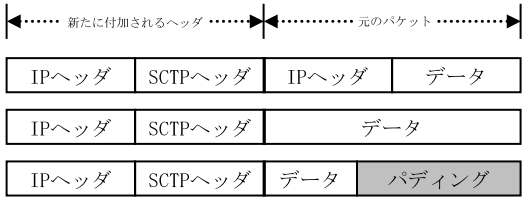

図 8　カプセル化されたゲートウェイ間のパケット
Fig. 8　Encapsulation for packets between gateways.

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．なお，同一の鍵で同一の平文を暗号化すると，つねに同一の暗号文となる ECB（Electronic CodeBook）モードは使用しない．送信間隔はパフォーマンスへの影響を考慮し，バッファに 50%以上の空き容量がある場合にのみ 0〜10 ms のランダムな遅延を付加する．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 におけるストリームの数は 1 とし，実装には lksctp1.0.9 [3] を用いた．SCTP を選択した場合のカプセル化されたパケットの一例を**図 8** に示す．この例では，元のパケットは 3 つに分割され，各々に新たな IP および SCTP ヘッダが付加されている．

6. 検 証 実 験

この章では，実験ネットワークにおいて検証実験を実施し，実装したトラフィックパターンを隠すアプリケーションベース VPN の性能を評価する．

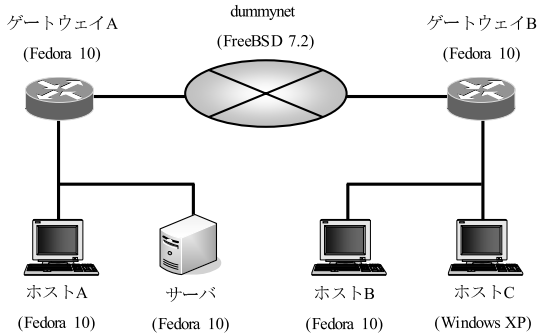

図 9　実験ネットワーク
Fig. 9　An experimental network.

6.1 実 験 環 境

図 9 に示す実験ネットワークを利用して実装したアプリケーションを検証する．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．ルータであるゲートウェイ A およびゲートウェイ B で実装したアプリケーションを動作させ，この間を FreeBSD に標準で組み込まれている dummynet [1] を介して VPN を構成する．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．サーバでは電子メール，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，ホスト A〜C からこれらのサービスを利用させる．

6.2 実 験 内 容

6.2.1 RTT 計測

ping コマンドを利用し，SCTP を用いた場合のホスト B とサーバ間の RTT（Round Trip Time）を計測する．プロトコルを UDP，パケット長を 1,024 byte とした場合の RTT（59.5 ms）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．この実験における dummynet の設定は**表 3** に示すとおりである．

6.2.2 スループット計測

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 [2] を利用し，SCTP を用いた場合のホスト B とサーバ間のスループットを計測する．RTT の計測と同様に，プ

表 3　dummynet の設定
Table 3　Parameters of a dummynet.

	遅延時間	損失率
1	0	0
2	10 ms	0
3	30 ms	0
4	30 ms	5%

表 4　各サービスを提供するアプリケーション
Table 4　Applications that provide each service.

サービス	アプリケーション
電子メール	Postfix および Dovecot
DNS	BIND
Web	Apache
時刻同期	ntp
動画のストリーミング	vlc media player

ロトコルを UDP，パケット長を 1,024 byte とした場合のスループット（602.4 KB/s）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．この実験における dummynet の設定は表 3 に示すとおりである．

6.2.3　機 能 確 認

サーバで提供する電子メール，DNS，Web，時刻同期，動画のストリーミング等のサービスを，ホスト A〜C から問題なく利用できることを確認する．サーバで各サービスを提供するアプリケーションの名称は**表 4** のとおりである．ストリーミングには，ビットレート 256 kb/s，フレームレート 15 fps，オーディオビットレート 128 kb/s，サンプリング周波数 44.1 kHz の mp4 形式の動画ファイルを用いる．また，パケット長があらかじめ設定した一定の値となっていることを確認するために，傍受したパケットを文献 43) で提案されているパケット長を時系列にプロットする手法で可視化する．

6.3　実 験 結 果

6.3.1　RTT 計測

SCTP における各パケット長ごとの RTT の相対値を**図 10** 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 はパケット長にかか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

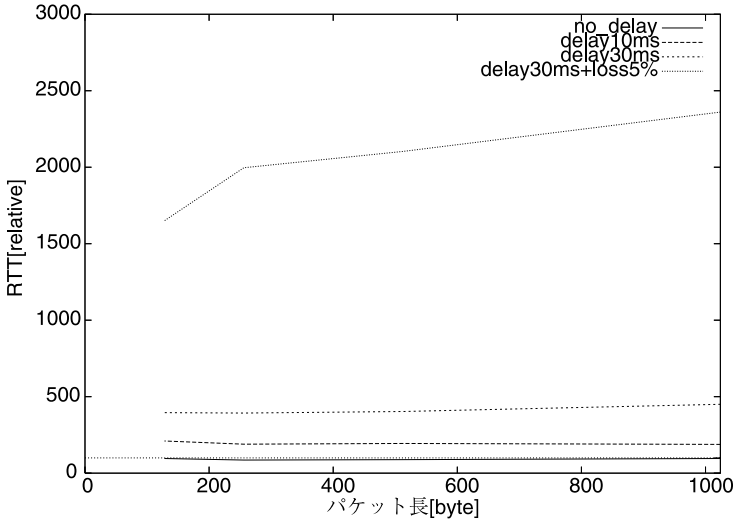

図 10　パケット長ごとの RTT の相対値
Fig. 10　Relative values of RTT.

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．また，RTT のデータ全体の平均値に対する絶対偏差の平均は 9.3 ms となった．これは，ファイル転送時等で連続した大容量のパケットを送信しない場合には，バッファに 50%以上の空き容量がある状態が保持され，0〜10 ms の遅延がランダムで付加された効果であると考えられる．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，パケットの損失にともない，再送要求が発生したためであると考えられる．

6.3.2　スループット計測

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 11** 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 256〜1,024 byte に設定した場合のスループットは基準値の 0.9〜1.4 倍の値となった．パケット長を 128 byte に設定した場合には，スループットは基準値の 0.2〜0.6 倍の値に大きく低下した．この原因は，パケット長を短くすることにより分割数が増大し，実装がボトルネックとなってグットプットが低下したためであると考えられる．この結果から，パケット長を 256 byte 以上にすれば，UDP の場合に近いスループットが期待できることが確認できた．

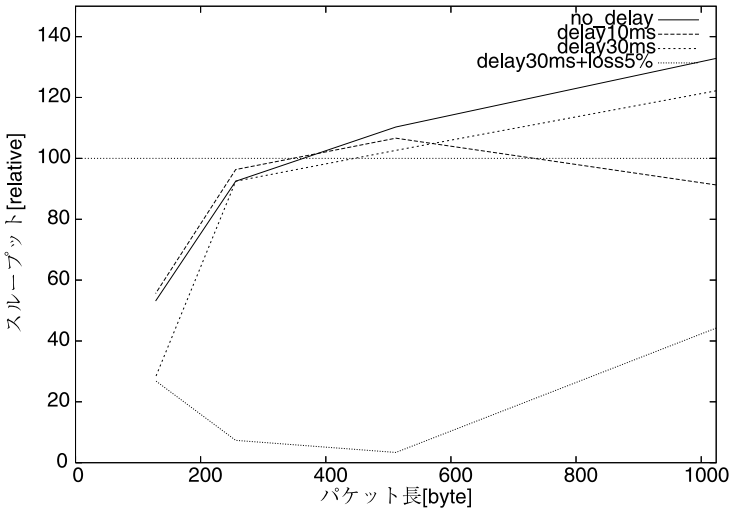

図 11　パケット長ごとのスループットの相対値
Fig. 11　Relative values of throughput.

表 5　機能確認の結果
Table 5　Results of a functional test.

プロトコル	アプリケーション	確認項目	確認結果
ICMP	ping コマンド	応答が得られること	○
SMTP，POP，IMAP	thunderbird	電子メールの送受信ができること	○
DNS	nslookup コマンド	名前解決および逆引きができること	○
HTTP，HTTPS	firefox	Web ページが表示できること	○
NTP	ntpdate コマンド	時刻同期ができること	○
RTP，RTCP	vlc media player	動画の視聴ができること	△

5%の損失率を与えた場合にはパケット長にかかわらずスループットは不安定となり，基準値の 0.5 倍に満たない低い値となった．これは，RTT 計測と同じく，パケットの損失による再送によりスループットが低下したことによるものであると考えられる．

6.3.3　機 能 確 認

表 5 に各サービスを提供するプロトコルごとの機能確認の結果を示す．表中のアプリケーションは機能確認のためにクライアントで用いたアプリケーションの名称であり，確認項目には具体的に確認した事項を示している．また，サーバのログおよびクライアント（ホスト B または C）のコンソールを確認し，エラー等が発生していないことも確認した．この結果，電子メール，DNS，Web，時刻同期等のプロトコルが，実装したトラフィックパターンを隠すアプリケーションベース VPN を介して問題なく動作することを確認することができた．しかしながら，動画のストリーミングに関しては，クライアントで再生することはできたが，十分な速度で視聴することはできなかった．この原因は，RTP には揺らぎの補正機能があることおよび，実験結果から RTT の分散はそれほど大きくなかったことから，スループットが十分ではないためであると考えられる．

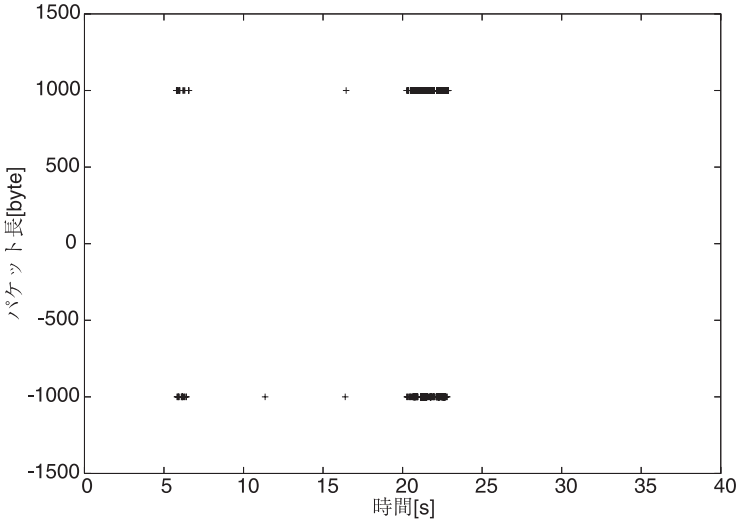

図 12　傍受した通信のパケット長時系列表示
Fig. 12　A time-packet sizes graph of eavesdropped communication.

次に，傍受した通信を文献 43) で提案されている手法で可視化した結果を**図 12** に示す．図中の縦軸は傍受したパケットの長さ，横軸はそのパケットを傍受した時間を表し，ここでは監視開始時刻からの経過時間を示す．縦軸の正の値はゲートウェイ A 側からゲートウェイ B 側への送信を表し，負の値はゲートウェイ B 側からゲートウェイ A 側への送信を表す．ゲートウェイのパケット長は 1,024 byte に設定されており，長さが 0 byte である制御パケットは除外している．図 12 から，パケット長があらかじめ設定した一定の値となっていることが確認できる．

7. 考　　察

7.1 秘 匿 性

本論文で実装したトラフィックパターンを隠すアプリケーションベース VPN は，均一の長さのパケットのみ送信する．したがって，単純に最初の数個のパケット長のみを利用した高速で高精度なアプリケーション推定手法[5] は完全に無力化することができる．また，機械学習を用いる手法のうち，パケット長のみを用いる手法[23] を無力化することができる．パケット長や方向等の属性値の遷移パターンを抽出する手法[21),41] に関しては，すべてのパケット長が均一となり，遷移パターンが発生しなくなるため，無力化することができる．

パケット長のほかにタイミングを利用する手法[22] に関しては，完全に無力化することはできない．しかしながら，実装したアプリケーションベース VPN は，乱数を使用してランダムな遅延を付加することによってタイミングを変更する対策を採用している．また，タイミングのみによってアプリケーションやプロトコルを推定する手法は提案されていない．たとえば，文献 39) で示されたパケット到着間隔に基づく分析手法では，リアルタイムトラフィックの抽出を実現しているが，個々のアプリケーションの推定には至っていない．したがって，これらの手法に対しても，アプリケーションやプロトコルの推定を妨げるある程度の効果があるものと考えられる．その他の機械学習を用いる手法[23),32),34] に関しても同様に，完全に無力化することはできないものの，分析の対象となるパケット長が均一となり，タイミングも乱数で変化するため，ある程度の検出を妨げる効果があることが期待できる．

7.2 従来の VPN との違い

OpenSSH，OpenVPN 等の既存のアプリケーションベース VPN は，暗号により通信の秘匿性を保証する．しかしながら，これらのアプリケーションベース VPN はトラフィックパターンをほとんど変えずにパケットを中継するため，挙動分析により情報システムの脆弱性に関する情報が漏洩してしまう可能性がある．よって，既存のアプリケーションベース VPN は挙動分析に対して脆弱である．

既存の挙動分析対策を施した VPN としては，IPsec[19] において TFC パディングが提案されている[20]．提案方式と IPsec（TFC パディング）の違いを**表 6** に示す．提案方式では，インターネット等の信頼性の低いネットワークを通じて DNS，NTP，SNMP 等の UDP を利用するアプリケーションの信頼性を向上させることができる．また，SCTP は TCP と同様にコネクションベースのプロトコルであるため，ネットワークの構成や設定を変更せずに，NAT（Network Address Translation）ルータを透過して VPN を構築することも可能である．既存の IPsec（TFC パディング）では，UDP 等のデータ伝送の信頼性を保証しないプロトコルを利用するアプリケーションに十分な信頼性を提供したり，単体で NAT ルータを透過して VPN を構築したりする機能を持たない．さらに，IPsec（TFC パディング）ではパケット長を短くする場合に，IPv6 の分割機能をそのまま活用している．そのため，攻撃者はパケットを再構築し，元のトラフィックパターンの推定を試みることが可能である．提案方式では，分割機能に用いられるフィールドも暗号化されるため，攻撃者はパケットを再構築することはできない．しかしながら，提案方式はアプリケーションベース VPN であるためオーバヘッドは高めであり，TCP over TCP の問題も発生する可能性がある．したがって，高いパフォーマンスが要求される場合には，提案方式は適しているとはいえない．提案方式は，高いパフォーマンスではなく，データ伝送の信頼性や NAT ルータを透過する汎用性が求められる場合に有効な方式であるといえる．このように，本研究で試作した SCTP を用いたアプリケーションベース VPN は，IPsec 等の信頼性を提供しない VPN とは異なる役割を演じるものである．

表 6　IPsec（TFC パディング）との違い
Table 6　Difference with IPsec (TFC Padding).

	提案方式	IPsec（TFC パディング）
データ伝送の信頼性	あり	なし
NAT ルータの透過	単体で可能	単体では不能
IPid によるパケットの復元	不能	可能
オーバヘッド	高い	低い

7.3 効　　果

挙動分析対策を実装するためには，パケット長とそのタイミングを制御する必要がある．本研究で提案した SCTP を用いたアプリケーションベース VPN の実現方式は，パケット長を厳密に制御することが可能であり，信頼性が保証されるという SCTP の特長を用いている．これらの特長は，本研究で示したアプリケーションベース VPN の実現方式以外にも，あらゆるネットワークアプリケーションにおける挙動分析対策の実装に適用することが可能であると考える．ネットワークアプリケーションにおいて，SCTP を用いずにパケット長の制御と信頼性の確保を実現しようとする場合，プロトコルを実装するのと同等の作業量が必要となる可能性があり，開発の効率は著しく低下する．本研究で提案した実現方式により，効率良くパケット長を制御し，トラフィックパターンを隠すネットワークアプリケーションを実装することが可能となる．

また，既存の VPN 以外のネットワークアプリケーションについても，挙動分析に対する脆弱性がないとは限らない．たとえば，挙動分析は IP 電話の使用言語の推定[42] にも応用されている．挙動分析により IP 電話の通話内容を推定することが可能となれば，重大な脅威となることは想像しにくくない．今後，通常のネットワークアプリケーションを実装する場合にも，挙動分析対策を検討する必要性が生じる可能性は否定できない．

8. おわりに

本論文では，パケット長と送信間隔を変更することにより，トラフィックパターンを隠すアプリケーションベース VPN を試作し，実験によりその特徴を分析するとともに，実装における課題を明らかにした．さらに，実装における課題を解決するために，SCTP を用いたアプリケーションベース VPN の実現方式を提案し，その性能を検証実験により評価した．

挙動分析対策を施すことで，攻撃者による情報システムの内部で稼動するホストの OS やアプリケーションの推定を妨げることができる．これにより，情報システムの内部の脆弱性が外部に知られることを防ぎ，結果として情報システムのセキュリティを向上させることができる．また，ネットワークの実装に起因する致命的な脆弱性があった場合にも，その脆弱性の存在を隠し，通信内容を秘匿することができる可能性を高めることができる．このように，挙動分析対策は，暗号通信の秘匿性を向上させる効果があるものと考えられる．

本研究では，Low Latency システムにおいて 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を制御することができるアプリケーションベース VPN を実現した．しかしながら，実装したアプリケーションベース VPN の性能は十分ではない．動画のストリーミング再生等を提供するリアルタイム性の高いアプリケーションが実用的に利用できるように，スループットを向上させることが今後の課題である．

また，本研究ではパケット長を均一とし，送信間隔をランダムにする手法を採用し，どのようにしてパケット長とタイミングを制御するアプリケーションベース VPN を実装するのかを示した．しかしながら，どのようにパケット長とタイミングを変更すれば，効率良く秘匿性を向上させることができるかという検討は十分ではない．分析手法に対する変更手法の秘匿効果と通信効率の関係については，定量的に評価する必要がある．採用したパケット長を固定長に変更する方式は，分析手法に対しては高い秘匿効果が期待できるが，あまり効率が良いとはいえない．効率を向上させるためには，極力元のトラフィックパターンを保持する必要があるが，元のトラフィックパターンを保持することで秘匿性は低下する．このように，挙動分析対策の効果と通信効率はトレードオフの関係にあり，いかに効率的に挙動分析対策を実現するかは大きな課題である．

参考文献

1) dummynet, available from ⟨http://info.iet.unipi.it/~luigi/dummynet/⟩.
2) Iperf, available from ⟨http://sourceforge.net/projects/iperf/⟩.
3) The Linux Kernel Stream Control Transmission Protocol, available from ⟨http://lksctp.sourceforge.net/⟩.
4) Metasploit Framework, available from ⟨http://www.metasploit.com/projects/Framework/⟩.
5) 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).
6) Berthold, O., Federrath, H. and Kopsell, S.: Web MIXes: A System for Anonymous and Unobservable Internet Access, *Proc. International Workshop on Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability*, pp.115–129 (2001).
7) Berthold, O. and Langos, H.: Dummy Traffic Against Long-term Intersection Attacks, *Proc. 2nd International Workshop on Privacy Enhancing Technologies*, Vol.2482, pp.110–128 (2002).
8) Braden, R., Zhang, L., Berson, S., Herzog, S. and Jamin, S.: Resource Reservation Protocol, RFC 2205 (1997).
9) Chaum, D.L.: Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms, *Comm. ACM*, Vol.24, No.2, pp.84–88 (1981).
10) Danezis, G., Dingledine, R. and Mathewson, N.: Mixminion: Design of a Type III Anonymous Remailer Protocol, *Proc. 2003 IEEE Symposium on Security and Privacy*, pp.2–15 (2003).
11) Dingledine, R. and Mathewson, N.: Tor: The Second-Generation Onion Router, *Proc. 13th USENIX Security Symposium*, pp.303–320 (2004).
12) Dusi, M., Crotti, M., Gringoli, F. and Salgarelli, L.: Tunnel Hunter: Detecting Application-Layer Tunnels with Statistical Fingerprinting, *Computer Networks* (*COMNET*), Vol.53, No.1, pp.81–97 (2009).
13) Ferguson, N., Schroeppel, R. and Whiting, D.: A Simple Algebraic Representation of Rijndael, *Revised Papers from the 8th Annual International Workshop on Selected Areas in Cryptography*, pp.103–111 (2001).
14) Fu, X., Graham, B., Bettati, R. and Zhao, W.: Analytical and Empirical Analysis of Countermeasures to Traffic Analysis Attacks, *Proc. 2003 International Confer-*

ence on Parallel Processing, pp.483–492 (2003).
15) Fu, X., Graham, B., Bettati, R. and Zhao, W.: On Countermeasures to Traffic Analysis Attacks, *Proc. 4th Annual IEEE Information Assurance Workshop*, pp.113–126 (2003).
16) Gebsk, M., Penev, A. and Wong, R.K.: Protocol Identification of Encrypted Network Traffic, *Proc. 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pp.49–54 (2006).
17) Gulcu, C. and Tsudik, G.: Mixing E-mail with Babel, *Proc. Symposium on Network and Distributed System Security*, pp.2–16 (2006).
18) 本田　治，大崎博之，今瀬　真，石塚美加，村山純一：TCP over TCP の性能評価：TCP トンネルがエンド–エンドのスループットおよび遅延に与える影響，信学技報 IN2004-121，Vol.104, No.438, pp.79–84 (2004).
19) Kent, S. and Seo, K.: Security Architecture for IP, RFC 4301 (2005).
20) Kiraly, C., Teofili, S., Bianchi, G., Cigno, R.L., Nardelli, M. and Delzeri, E.: Traffic Flow Confidentiality in IPsec: Protocol and Implementation, *The Future of Identity in the Information Society*, Vol.262, pp.311–324 (2008).
21) 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).
22) 北村　強，静野隆之，岡部稔哉：フロー挙動に基づくアプリケーション識別技術の開発，信学技報 NS2006-145，Vol.106, No.418, pp.35–38 (2006).
23) Kohara, M., Hori, Y., Sakurai, K., Lee, H. and Ryou, J.-C.: Flow Traffic Classification with Support Vector Machine by using Payload Length, *The 1st International Workshop on Network Traffic Control, Analysis and Applications* (*NTCAA 2009*) (2009).
24) Kohler, E., Handley, M. and Floyd, S.: Datagram Congestion Control Protocol, RFC 4340 (2006).
25) Levine, B.N., Reiter, M.K., Wang, C. and Wright, M.: Timing Attacks in Low-Latency Mix Systems, *Proc. 8th International Conference on Financial Cryptography*, Vol.3110, pp.251–265 (2004).
26) Liberatore, M. and Levine, B.N.: Inferring the Source of Encrypted HTTP Connections, *Proc. 13th ACM Conference on Computer and Communications Security*, pp.255–263 (2006).
27) Matsumoto, M. and Nishimura, T.: Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator, *ACM Trans. Modeling and Computer Simulation*, Vol.8, No.1, pp.3–30 (1998).
28) Mimura, M. and Nakamura, Y.: A Tunnel that Conceals Packet's Behavior Against Traffic Analysis Attack, *Memoirs of the National Defense Academy*, Vol.49, No.2, pp.1–19 (2010).
29) Mimura, M. and Tanaka, H.: Behavior Shaver: An Application Based Layer 3 VPN that Conceals Traffic Patterns Using SCTP, *Proc. 2010 International Conference on Broadband, Wireless Computing, Communication and Applications* (*BWCCA 2010*), pp.666–671 (2010).
30) 南浦優樹，阿多信吾，中村信之，中平佳裕，村田正幸，岡　育生：暗号化・平文トラヒックの差分分析とその特徴にもとづく暗号化トラヒック検出法，信学技報 ICM2008-89, Vol.108, No.481, pp.179–184 (2008).
31) Moeller, U., Cottrell, L., Palfrader, P. and Sassaman, L.: Mixmaster Protocol Version 2, RFC 2026 (2003).
32) Moore, A.W. and Zuev, D.: Internet Traffic Classification Using Bayesian Analysis Techniques, *The International Conference on Measurement and Modeling of Computer Systems 2005* (2005).
33) Nagle, J.: Congestion Control in IP/TCP Internetworks, RFC 896 (1984).
34) Nguyen, T. and Armitage, G.: A Survey of Techniques for Internet Traffic Classification using Machine Learning, *IEEE Communications Surveys and Tutorials*, Vol.10, No.4, pp.56–76 (2008).
35) Reed, M.G., Syverson, P.F. and Goldschlag, D.M.: Anonymous Connections and Onion Routing, *IEEE Journal Selected Areas in Communications*, Vol.16, pp.482–494 (1998).
36) Reiter, M.K. and Rubin, A.D.: Crowds: Anonymity for Web Transactions, *ACM Trans. Information and System Security*, Vol.1, pp.66–92 (1998).
37) Shmatikov, V. and Wang, M.-H.: Timing Analysis in Low-latency Mix Networks: Attacks and Defenses, *Proc. 11th European Symposium on Research in Computer Security* (*ESORICS*), Vol.4189, pp.18–33 (2006).
38) Stewart, R.: Stream Control Transmission Protocol, RFC 4960 (2007).
39) Tai, M., Ata, S. and Oka, I.: Environment-Independent Online Real-Time Traffic Identification, *Proc. 4th International Conference on Networking and Services*, pp.230–235 (2008).
40) Timmerman, B.: Secure Dynamic Adaptive Traffic Masking, *Proc. 1999 Workshop on New Security Paradigms*, pp.13–24 (1999).
41) 和泉勇治，阿部康一，根元義章：メッセージの遷移パターンに基づくネットワークアプリケーション識別システムの試作，電子情報通信学会論文誌 D，Vol.J93-D, No.10, pp.2257–2267 (2010).
42) Wright, C.V., Ballard, L., Monrose, F. and Masson, G.M.: Language Identification of Encrypted VoIP Traffic, *Proc. 16th Annual USENIX Security Symposium* (2007).
43) Wright, C.V., Monrose, F. and Masson, G.M.: Using Visual Motifs to Classify Encrypted Traffic, *Proc. 3rd International Workshop on Visualization for Computer*

Security, pp.41–50 (2006).
44) Fu, X., Graham, B., Bettati, R. and Zhao, W.: On Effectiveness of Link Padding for Statistical Traffic Analysis Attacks, *Proc. 23rd International Conference on Distributed Computing Systems*, pp.340–347 (2003).
45) 山田　明，三宅　優，寺邊正大，橋本和夫：SSL/TLS で暗号化された Web 通信に対する侵入検知システム，情報処理学会論文誌，Vol.49, No.3, pp.1144–1154 (2008).
46) Zhu, Y., Fu, X., Graham, B., Bettati, R. and Zhao, W.: On Flow Correlation Attacks and Countermeasures in Mix Networks, *Proc. 4th International Workshop on Privacy-Enhancing Technologies*, Vol.3424, pp.207–225 (2004).

(平成 22 年 12 月 15 日受付)
(平成 23 年 6 月 3 日採録)

三村　守（正会員）

2001 年防衛大学校情報工学科卒業．同年海上自衛隊入隊．2008 年防衛大学校理工学研究科前期課程修了．同年海上自衛隊保全監査隊勤務（現職）．2011 年情報セキュリティ大学院大学博士後期課程修了．ネットワークセキュリティに興味を持つ．博士（情報学）．

田中　英彦（名誉会員）

1970 年東京大学大学院工学系研究科電気工学専門課程修了．工学博士．東京大学にて計算機アーキテクチャ，並列処理，人工知能，自然言語処理，分散処理，メディア処理等の教育・研究に従事．東京大学工学部教授，同大学院情報理工学系研究科長を経て，2004 年情報セキュリティ大学院大学情報セキュリティ研究科長・教授．情報処理学会名誉員，人工知能学会論文賞，ACM SIGGRAPH'99 Impact Paper Award，人工知能学会功績賞，東京都科学技術功労者表彰，経済産業大臣表彰等受賞．情報・システム研究機構教育研究評議会評議員，日本学術会議会員，IEEE Fellow，東京大学名誉教授．