知能と複雑系　　125−10
（２００１．７．２４）

# QoS を考慮したインターネットセキュリティプロトコルの提案

福田 洋治† 佐野 工† 白石 善明‡ 森井 昌克†

† 徳島大学 工学部 知能情報工学科
〒 770–8506 徳島市南常三島町 2-1
TEL/FAX: 088–656–7487
E-mail: {youji, takumi, morii}@is.tokushima-u.ac.jp

‡ (有) ナオゼンネットワークス
〒 771–0125 徳島市川内町金岡 40-1
TEL/FAX: 088–637–1268
E-mail: zenmei@naozen.co.jp

あらまし　ネットワークのサービス品質 (Quality of Service; QoS) を保証する QoS 制御技術は，ネットワーク・ポリシーに基づいてトラヒックをいくつかのクラスに分類してそれぞれ異なった扱い方をする，差別化されたサービスクラスを実現する．したがって，QoS 制御に利用されるパケットの識別情報 (IP ヘッダや TCP/UDP ヘッダの情報) は，ネットワーク上で閲覧できなければならない．しかし，ESP(Encapsulating Security Payload) や SSL(Secure Socket Layer) などのアプリケーション層に属さない，暗号化をともなうセキュリティ・プロトコルは，元パケットの識別情報を秘匿または隠蔽し，ネットワーク上の QoS 制御を妨げてしまう．そこで本稿では，ネットワーク上の QoS 制御を考慮したセキュリティ・プロトコル (ESP considered QoS, ESPQ) を提案し，その安全性と有効性を検討する．

キーワード　IPSec, ESP, QoS, 機密性, 完全性

# Proposal of Internet Security Protocol considered QoS

Youji FUKUTA† Takumi SANO† Yoshiaki SHIRAISHI‡ Masakatu MORII†

† The University of Tokushima
Tokushima, 770–8506, Japan
TEL/FAX: 088–656–7487
E-mail: {youji, takumi, morii}@is.tokushima-u.ac.jp

‡ NaozenNetworks Co.,Ltd.
Tokushima, 771–0125, Japan
TEL/FAX: 088–637–1268
E-mail: zenmei@naozen.co.jp

Abstract　Quality of Service(QoS) control is applied to data packets so that we can distinguish themselves by service classes. On the other hand, many Virtual Private Network(VPN) solutions which provide confidentiality of a packet by encryption, and hide the information which need to distinguish the service class. Therefore, we can not control QoS of packets made by existing VPN solutions. In this paper, we propose a new security protocol, Encapsulating Security Payload considered QoS(ESPQ) improved Encapsulating Security Payload(ESP) which has been already defined as a security protocol at the network layer in IP Security Protocol(IPSec). The protocol does not disturb QoS control, and can provide a packet with confidentiality and integrity like ESP.

Key words　IPSec, ESP, QoS, Confidentiality, Integrity

# 1 はじめに

従来，企業や公共機関の拠点間の通信，すなわち組織のプライベートネットワーク同士の相互接続には，第三者に情報を盗み見されない，あるいは情報が偽造，改竄されることなく受信側に届くという意味の信頼性のある通信を行うために，専用回線が用いられている場合が多かった．近年は，組織のネットワーク運用コストを削減するために，専用回線の代りにインターネットを用いたネットワーク相互接続の需要が高まっている．

インターネットのような公共のネットワークを用いて，プライベートネットワーク同士の相互接続 (network-to-network interconnection) や遠隔地のホストから組織のプライベートネットワークに所属するホストへのアクセス (remote access) を仮想的に実現する技術として VPN(Virtual Private Network) がある．例えば，代表的な VPN ソリューションである，SSL(Secure Socket Layer) [20] や IPSec(IP Security protocol) [1] の中で定義されている ESP(Encapsulating Security Payload) [3] は，それぞれ TCP 層と IP 層のレベルでパケットに対して完全性 (integrity) と機密性 (confidentiality) を提供する セキュリティ・プロトコルとして，現在，広く利用されている．ここで，完全性とはデータや情報が正当，正確かつ完全に維持されていることを意味し，機密性とは第三者に対してデータや情報が秘匿されていることを意味する．

一方，近年の通信ネットワーク関連技術の進歩に伴って，多種多様な情報をリアルタイムに送受信するマルチメディア通信サービスが急速に普及している．実時間性を必要とするマルチメディア通信サービスは，情報を寸断することなく，また安定して提供されなければならない．したがって，実時間性や広帯域性，信頼性といったネットワークのサービス品質 (Quality of Service; QoS) を保証する QoS 制御技術はネットワーク提供者やネットワーク利用者の間で重要視されている [16]．しかし，前述した SSL や ESP に代表されるアプリケーション層よりも下位の層で実現される，暗号化を伴ったセキュリティ・プロトコルは，QoS 制御を行なう際に必要な各パケットの識別情報 (IP ヘッダの情報や，TCP/UDP ヘッダの情報) を秘匿または隠蔽するので，その結果として QoS 制御を妨げてしまう．

本稿では，通信の安全性を維持しながら，バックボーン・ネットワーク内あるいは組織ネットワーク内のルータ等の制御機器による QoS 制御を妨げないセキュリティ・プロトコルを提案し，その安全性を検討する．

# 2 セキュリティプロトコルと QoS

## 2.1 IPSec(IP Security protocol)

IPSec は，IPv4 および IPv6 の両方の環境において，既に安全性が評価されている標準的な暗号/認証アルゴリズムを用いて IP 層のトラヒックに様々なセキュリティ・サービスを提供する枠組みである．IPSec が提供するセキュリティ・サービスには，(1) アクセス制御，(2) データ送信元認証，(3) リプレイに対する保護，(4) 完全性の保証，(5) 機密性の保証が含まれる [1, 2, 3]．ここで，アクセス制御とは許可されていないエンティティの情報資源やネットワーク資源の不正利用を防ぐことであり，リプレイに対する保護とは正規のホストから送信されたパケットを使用した，なりすまし攻撃から受信側のホストを保護することである．

また，IPSec のセキュリティ・アーキテクチャの主要な構成要素を次に示す．

- セキュリティ・プロトコル（AH [2]，ESP [3]）
- SA(Security Associations) [1]
- 鍵管理 [4]
- 認証および暗号化のためのアルゴリズム [5, 6, 8, 9]

IPSec は，上記の構成要素をそれぞれ機能単位として扱い，その構成・使用方法のみを規定する．各構成要素の詳細は別の RFC の中で定義されている．

## 2.2 セキュリティ・プロトコル (ESP, AH)

IPSec では，AH(Authentication Header) と ESP(Encapsulating Security Payload) の 2 つのセキュリティ・プロトコルが定義されている．

AH は，2.1 節で述べたセキュリティ・サービスのうち，アクセス制御，完全性の保証，データ送信元認証，リプレイに対する保護 (オプション) を提供する．ESP は，AH が提供するセキュリティ・サービスに加えて，機密性の保証を提供する．

AH と ESP は，トンネルモード (Tunneling Mode) とトランスポートモード (Transport Mode) の 2 つのモードを実現する仕組みを持っている．トランスポートモードは，Host-to-Host の通信を行う場合に用いる方法であり，トンネルモードは，Network-to-Network の通信，および Network-to-Host の通信を行う場合に用いられる方法である．

また，IPSec をサポートするホスト (セキュリティ・ゲートウェイ) 同士で事前に共有しなければならない，セキュリティ・プロトコルや認証アルゴリズム，暗号化アルゴリズム，認証鍵や暗号化鍵などの情報は，Security Associations(SA) の中で指定される．

SA は，送信側と受信側の関係から成る単方向コネクションの，セキュリティ・ポリシーを規定する情報や暗号化/認証に必要な秘密情報を保持する情報レコード，あるいはそれを示す概念である [1]．SA は，セキュリティ・プロトコル，宛先 IP アドレス，SPI(Security Parameter Index) によって一意に識別される．

## 2.3 QoS 制御を妨げる暗号化通信

QoS 制御は，ネットワーク・ポリシーに従ってトラヒックをクラスに分類して，あるクラスに所属するトラヒックを他のクラスに所属するトラヒックと区別して扱う，差別化されたサービスクラス (differentiated Class of Service) を実現する [16]．

差別化されたサービスクラスを実現するためには，ネットワークの入口およびネットワークの内部ノードにおいて，

一般に 2 段階の過程を経る必要がある．ひとつはネットワークに入力されたトラヒックを識別する過程であり，もうひとつは入力されたトラヒックを適切に分類しネットワーク・ポリシーに合致するように整形する過程である．

前者において，差別化を行う際には何らかの判断基準に従ってトラヒックを識別する必要がある．一般的なトラヒックの差別化は，次に示す基準に基づいてトラヒックの識別と分類が行われる．

- プロトコル
- 送信元ポート番号
- 宛先ポート番号
- 送信元ホストアドレス
- 宛先ホストアドレス
- ソースデバイスインタフェース
- フロー

しかし，上記のパケットの識別情報は，暗号化を伴うセキュリティ・プロトコルによって，通常，秘匿または隠蔽されてしまう．例えば，ESP のような IP 層上で実現されるセキュリティ・プロトコルは，パケットに含まれるネットワーク層やトランスポート層のヘッダ情報を暗号化する．また，SSL のような TCP 層上で実現されるセキュリティ・プロトコルは，ポート・フォワーディングやトンネリングによって，パケット本来のサービスクラスを隠蔽してしまう．したがって，ESP や SSL は，明らかに QoS 制御を妨げていることがわかる．

そこで本研究では，QoS 制御を妨げないセキュリティ・プロトコルの提案を目的としている．

# 3 提案手法: ESPQ

## 3.1 ESPQ の特徴

本稿では，QoS 制御を妨げないセキュリティ・プロトコル ESPQ(ESP considered QoS) を提案する．ESPQ は，IP 層上のセキュリティ・プロトコル ESP を QoS 制御に関して拡張したセキュリティ・プロトコルである．

ESPQ は，通常の TCP/UDP パケットと同様に，元パケットの TCP/UDP ヘッダを暗号化することなく IP ヘッダの直後に配置する．したがって，ルータは ESPQ のパケットを通常の TCP/UDP パケットとして取り扱うことが可能であり，すなわち，ESPQ のパケットはネットワーク上で QoS 制御が可能である．

また，ESPQ は，アクセス制御，データ送信元認証，リプレイに対する保護，完全性の保証，機密性の保証の 5 つのセキュリティ・サービスも ESP と同様に提供する．

## 3.2 ESPQ パケットのフォーマット

ESPQ のトランスポートモードおよびトンネルモードのパケットフォーマットを図 1 に示す．また，IP ヘッダを除いた ESPQ パケットフォーマットを図 2 に示す．

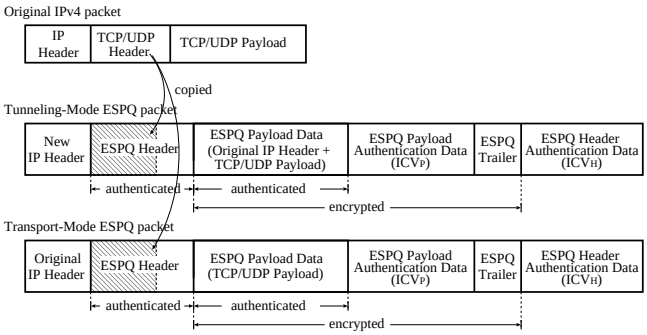


図 1: ESPQ パケット

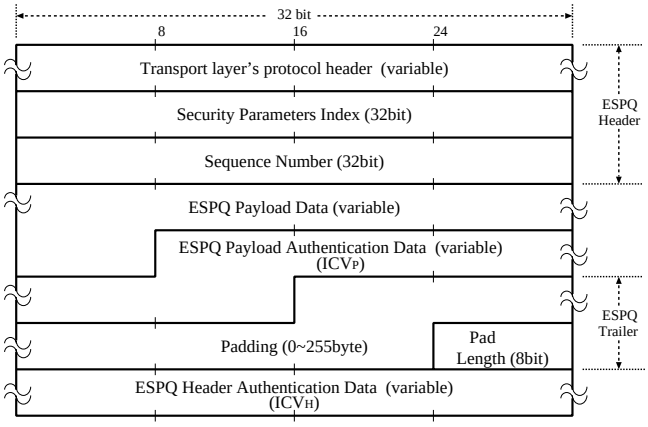


図 2: ESPQ のフォーマット

図 2（ESPQ パケットのフォーマット）の各フィールドについて説明する．

- Transport layer's protocol header

  このフィールドには，オリジナル IP パケットのトランスポート層のプロトコルヘッダ（TCP ヘッダ，UDP ヘッダ）がコピーされる．

- Security Parameters Index(SPI)

  SPI は，SA を一意に識別する 32 ビットの値である．ESPQ では，SPI と宛先 IP アドレスの組み合わによって SA を識別する．SPI は，SA の確立時に宛先システムによって選択される．

- Sequence Number

  この 32bit の領域には，パケットを送信する度に 1 ずつ増加されるカウンタ値 (シーケンス番号) が含まれる．シーケンス番号フィールドの処理は受信側に任せられる．送信側のカウンタと受信側のカウンタは、SA の確立時に 0 に初期化される．送信側のカウンタと受信側のカウンタは，ある SA 上で $2^{32}$ 番目のパケットを送信する前にリセットされなければならない．

- ESPQ Payload Data

  この領域は，元パケットの TCP/UDP ペイロード (トンネルモードの場合，元パケットの IP ヘッダも含ま

れる) を含む整数バイト長の可変長フィールドである．暗号化アルゴリズムが，暗号同期データ (IV など) を必要とする場合，そのデータはペイロード領域内で明示的に運ばれてもよい．

- ESPQ Payload Authentication Data

  この領域は，ESPQ ペイロードデータ領域のインテグリティチェック値 ($ICV_P$) を含む可変長領域である．この領域の長さは，認証アルゴリズムが生成するダイジェストの長さに依存する．

- Padding

  あるバイト数の，整数倍の長さの平文を要求する暗号化アルゴリズムが使用される場合，アルゴリズムが要求する長さまで平文を詰めるために，このパディング領域が使用される．特に，次に挙げるパディング長領域は，ESPQ ヘッダ認証データ領域が 4 バイトの境界で整列することを保証するために，4 バイトワード内で右側に寄るように整列されなければならない．パディングバイトは，符号なし 1 バイトの連続した整数値に初期化される．平文に追加される最初のパディングバイトには 1 と番号付けされ，それに続くパディングバイトでは単純に 2,3,4,$\cdots$ と増加していく．この領域の長さの範囲は，0 バイトから 255 バイトである．

- Pad Length

  この領域は，その直前のパディング領域のバイト長を示す 8 ビットの領域である．

- ESPQ Header Authentication Data

  ESPQ ヘッダ認証データ領域は，ESPQ ヘッダ領域のインテグリティチェック値 ($ICV_H$) を含む可変長領域である．この領域の長さは，認証アルゴリズムが生成するダイジェストの長さに依存する．

## 3.3 ESPQ のパケット処理

### 3.3.1 出力パケット処理: ESPQ パケットの構築

元パケットから ESPQ のパケットを構築する手続き，すなわち，出力パケット処理の手続きを以下に説明する．

**Step-A1** SA の検索

元パケットの宛先 IP アドレスから，それに対応する SA を検索する．

**Step-A2** ESPQ ヘッダの構築

元パケットの TCP/UDP ヘッダと Step-A1 で得られた SA の SPI とシーケンス番号から，ESPQ ヘッダを構築する．このとき，ESPQ ヘッダに含まれる TCP/UDP ヘッダのチェックサム領域は 0 に初期化される．

**Step-A3** ESPQ ペイロードデータ，$ICV_P$，ESPQ トレイラの作成

元パケットの TCP/UDP ペイロード (トンネルモードの場合，元パケットの IP ヘッダも含まれる) を ESPQ ペイロードデータを作成する．次に，認証アルゴリズムを用いて ESPQ ペイロードデータのインテグリティチェック値 ($ICV_P$) を計算し，ESPQ ペイロードデータの直後に付加する．続いて，ESPQ ペイロードデータと $ICV_P$ の結合領域に対して，ESPQ トレイラ (パディング領域とパディング長領域) を作成し，これを付加する．ここで使用する認証アルゴリズムは，SA が指定する．

**Step-A4** 暗号化

ESPQ ペイロードデータと $ICV_P$，ESPQ トレイラを暗号化アルゴリズムを用いて秘匿する．ここで使用する暗号化アルゴリズムは，SA が指定する．

**Step-A5** $ICV_H$ の作成と TCP/UDP ヘッダのチェックサムの計算

認証アルゴリズムを用いて ESPQ ヘッダのインテグリティチェック値 ($ICV_H$) を計算し，ESPQ トレイラの直後に付加する．次に，ESPQ ヘッダに含まれる TCP/UDP ヘッダのチェックサムを計算して，計算結果を TCP/UDP ヘッダのチェックサム領域に格納する．ここで使用する認証アルゴリズムは，SA が指定する．

**Step-A6** IP ヘッダの構築

ESPQ パケットの IP ヘッダを構築する．トランスポートモードの場合，元パケットの IP ヘッダを ESPQ パケットの IP ヘッダに用いる．トンネルモードの場合，送信元・送信先のゲートウェイの IP アドレスを含む新しい IP ヘッダを構築する．

**Step-A7** IP 分割

ESPQ パケットの長さが MTU(Maximum Transmission Unit) を越える場合，ESPQ パケットの構築後に IP 分割 (IP fragmentation) が行なわれる．ESPQ パケットはネットワークの配送経路上で分割される場合もあり，受信側では ESPQ パケットから元パケットの再構築を行う前に IP 分割されたパケットの再構築を行わなければならない．

### 3.3.2 入力パケット処理: 元パケットの再構築

ESPQ パケットから元パケットを再構築する手続き，すなわち，入力パケット処理の手続きを以下に説明する．

**Step-B1** IP 分割パケットの再構築

IP 分割されたパケットを受信した場合は，IP 分割パケットの再構築を行う．

**Step-B2** SA の検索

受信したパケットの，IP ヘッダのプロトコルの情報から，プロトコル (TCP または UDP) を判断する．次に，受信したパケットを ESPQ パケットと仮定して，ESPQ ヘッダの SPI 領域に対応する領域から値を取り出す．この値を用いて，SA を検索する．Step-B2 の

段階では，一般の TCP/UDP パケット と ESPQ パケットを区別できない．

**Step-B3** $ICV_H$ の検証

受信したパケットの TCP/UDP ヘッダのチェックサムを 0 に初期化する．次に，SA が指定する認証アルゴリズムを用いて，受信したパケットの，ESPQ ヘッダ領域に対応する領域のインテグリティチェック値$(ICV'_H)$ を計算する．続いて，$ICV'_H$ と受信したパケットの $ICV_H$ 領域に対応する領域から取り出した値を比較する．もし一致したならば，ESPQ パケットが受信された，と判断する．ただし，受信した ESPQ パケットの ESPQ ペイロードデータが偽造または改竄されている可能性が残っていることに注意する．

**Step-B4** 復号

復号アルゴリズムを用いて，ESPQ ペイロードデータと $ICV_P$ ，ESPQ トレイラの領域を復号する．ここで使用する復号アルゴリズムは，SA が指定する．

**Step-B5** $ICV_P$ の検証

復号された ESPQ ペイロードデータのインテグリティチェック値 $(ICV'_P)$ を計算する．次に，$ICV'_P$ と 復号した $ICV_P$ を比較する．もし一致したならば，正規の ESPQ パケットを偽造または改竄されることなく受信できた，と判断する．

**Step-B6** 元パケットの再構築

ESPQ ヘッダに含まれる TCP/UDP ヘッダと ESPQ ペイロードデータに含まれる TCP/UDP ペイロード (トンネルモードの場合，元パケットの IP ヘッダも含まれる) を再結合する．このとき，TCP/UDP ヘッダのチェックサムを計算する．次に，元パケットの IP ヘッダを構築する．トランスポートモードの場合，ESPQ パケットの IP ヘッダを元パケットの IP ヘッダに用いる．トンネルモードの場合，ESPQ ペイロードデータに元パケットの IP ヘッダが含まれているので，これを用いる．

### 3.4 パケットの認証と暗号化

この節では，ESPQ の認証と暗号化について説明する．

ESPQ は，ESPQ ヘッダと ESPQ ペイロードデータに対して次に示す認証および暗号化を行って，パケットの完全性と機密性を保証する．

ただし ESPQ では，認証および暗号化のアルゴリズムや SA 管理機構は ESP と同じものを使用している．図 3 は，ESPQ パケットの暗号化範囲と認証範囲を示したものである．

**［ESPQ ヘッダの認証］**

ESPQ は，次式を用いて ESPQ ヘッダを認証する．

$$ICV_H = A_1(M_H, K_A)$$

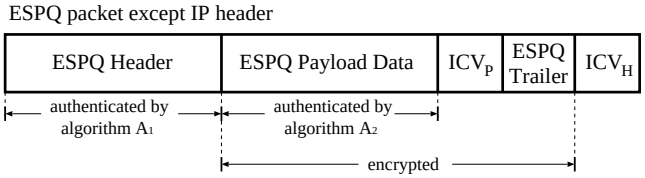


図 3: 暗号化範囲と認証範囲

ここで，認証される ESPQ ヘッダを $M_H$ ，認証鍵を $K_A$ ，鍵付きハッシュ関数を $A_1$ ，ESPQ ヘッダのインテグリティチェック値を $ICV_H$ としている．$A_1$ と $K_A$ は SA の中で指定され，IPSec の SA 管理機構によって安全に共有される．ESP と同様に $A_1$ には HMAC-MD5 [5] や HMAC-SHA [6] などの安全なハッシュ関数を用いる．

**［ESPQ ペイロードデータの認証と暗号化］**

ESPQ は，次式を用いて ESPQ ペイロードデータの認証および暗号化を行う．

$$ICV_P = A_2(M_P),$$
$$E(M_P||ICV_P||Tr, K_E)$$

ここで，認証される ESPQ ペイロードデータを $M_P$ ，ハッシュ関数を $A_2$ ，ESPQ ペイロードデータのインテグリティチェック値を $ICV_P$ ，ESPQ トレイラを $Tr$ ，暗号化関数を $E$ ，暗号化鍵を $K_E$ としている．$A_2$ と $E$ ，$K_E$ は SA の中で指定され，これらは IPSec の SA 管理機構によって安全に共有される．ESP と同様に $A_2$ には HMAC-MD5 や HMAC-SHA などの安全なハッシュ関数を用いる．また $E$ には，CBC モードや CFB モードの Triple-DES [13] や RC5 [14] を用いる．

ESPQ ペイロードデータの認証方法は，ハッシュ関数と暗号化関数を併用した認証方法である．したがって ESPQ では，認証と暗号化は共に必須である．また，この認証方法の安全性は，暗号化関数の安全性に依存する [18]．

## 4 実装と評価

### 4.1 実験環境

実験環境を図 4 に示す．ESPQ を実装した 2 台のセキュリティ・ゲートウェイを，それぞれ別の QoS ルータを介し共通の外側ネットワークに接続して，Network-to-Network の IP-VPN を構築する．各セキュリティ・ゲートウェイの内側ネットワークには，1 台ずつトラヒック測定用 PC が接続されている．また，各セキュリティ・ゲートウェイが接続されている QoS ルータには，ネットワークに負荷をかけるためのトラヒック測定用 PC がそれぞれ 6 台ずつ接続されている．

トラヒック測定用 PC では，測定ツール DBS(Distributed Benchmark System) Version 1.1.5 [19] が常駐している．

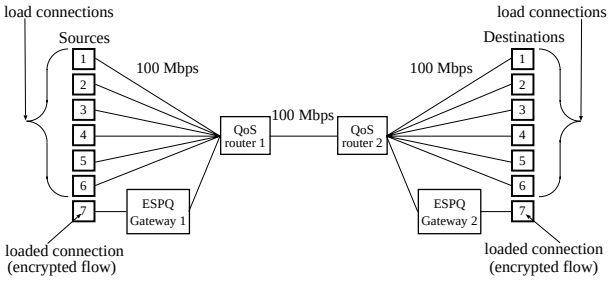


図 4: 実験環境

表 1: ESPQ Gateway 1,2 のマシンスペック

| | |
|---|---|
| OS | RedHat Linux-R6.2 |
| CPU | Celeron 667MHz |
| Memory | 128MB |
| NIC1 | RealTek RTL8139 Fast Ethernet |
| NIC2 | RealTek RTL8139 Fast Ethernet |
| 暗号化アルゴリズム | CBC モード Triple-DES |
| 認証アルゴリズム | HMAC-MD5 |

表 2: QoS router 1,2

| | |
|---|---|
| Router | CISCO 7204VXR |
| IOS | C7200-JS-M 12.0(7)T |

表 3: トラヒック測定用 PC のマシンスペック

| | |
|---|---|
| OS | FreeBSD 4.2R |
| CPU | Pentium II 450MHz |
| Memory | 128MB |
| NIC | 3Com 3c905B-TX Fast Etherlink XL |

DBS は，(1) 複数のホスト間の同時データ転送が可能，(2) 個々のパケットの送受信時刻を記録可能，という 2 つの特徴を持っている．したがって DBS は，ネットワークを輻輳させてパケットの喪失が起こる状況を作り出し，データストリームの，スループットの時刻変動を測定することが可能である．

ESPQ を実装したゲートウェイのスペックを表 1 に示す．また，QoS ルータと測定用 PC のマシンスペックをそれぞれ表 2 と表 3 に示す．

## 4.2 評価実験

本節では，ESPQ のパケットが QoS 制御可能であることを示す実験を行う．前節で説明した実験環境を使って次の 2 種類のトラヒック測定実験を行った．

- トラヒック測定実験 1

  各 QoS ルータに接続されている 6 台のトラヒック測定用 PC を使って，ネットワークに負荷をかけるための TCP トラヒック 1 (表 4) を 6 本発生させる．

  このとき，Network-to-Network の IP-VPN を構築するセキュリティ・ゲートウェイに接続された 双方 1 台ずつのトラヒック測定用 PC を使って，1 本の TCP トラヒック 2(表 5) を発生させる．このトラヒックは，ESPQ のセキュリティ・ゲートウェイを経由して認証・暗号化されたトラヒックである．

  負荷用のトラヒック 6 本と認証・暗号化されたトラヒック 1 本が同時に外側ネットワークを流れる状況で，QoS 制御を行わない場合のトラヒックのスループットをそれぞれのトラヒック測定用 PC で計測する．

  この時のスループットの推移を基準にして，トラヒック測定実験 2 の結果と比較を行う．

表 4: トラヒック 1

| | |
|---|---|
| プロトコル | TCP |
| 宛先ポート番号 | 50000 – 50005 |
| トラヒックパターン | 8,192 バイトのバルク転送 |
| グラフ中の表記 | load connection |

表 5: トラヒック 2

| | |
|---|---|
| プロトコル | TCP |
| 宛先ポート番号 | 50010 |
| トラヒックパターン | 8,192 バイトのバルク転送 |
| グラフ中の表記 | loaded connection |

- トラヒック測定実験 2

  トラヒック測定実験 1 と同様の 7 本のトラヒック (負荷用 の TCP トラヒック 6 本と認証・暗号化された TCP トラヒック 1 本) を発生させる．このとき，外側ネットワーク上の QoS ルータで TCP ヘッダの宛先サービスポート番号を用いて QoS 制御を行う．

  外側ネットワーク上で行う QoS 制御は，TCP ヘッダの宛先ポート番号が 50000 番 から 50005 番までのトラヒック (表 4) を非優先トラヒック，50010 番のトラヒック (表 5) を優先トラヒックとして扱い，それぞれの優先度 (非優先は “0”，優先は “5” とする) を IP ヘッダの TOS 領域中にある IP 優先度 (IP Precedence) 領域に格納して WFQ(Weighted Fair Queueing) を行う，というものである．(IP ヘッダの TOS 領域の書換えは，外側ネットワークの入口にある QoS ルータが行う．WFQ による優先制御は，外側ネットワークの入口の QoS ルータおよび内部ノードで行われる．)

**[実験結果]**

トラヒック測定実験 1 と トラヒック測定実験 2 の結果を図 5 と図 6 に示す．

ESPQ ゲートウェイの処理能力はスループットにして約 15 Mbps である．図 5 では，トラヒック 2 はトラヒック 1 により圧迫されている．一方，図 6 では，パケットの優先制御が行われているので，トラヒック 2 が優先的に取り扱われている．このように，トラヒック 2 は，ESPQ によって認証・暗号化されているにも関わらず，ネットワーク上で 容易に QoS 制御が可能であ

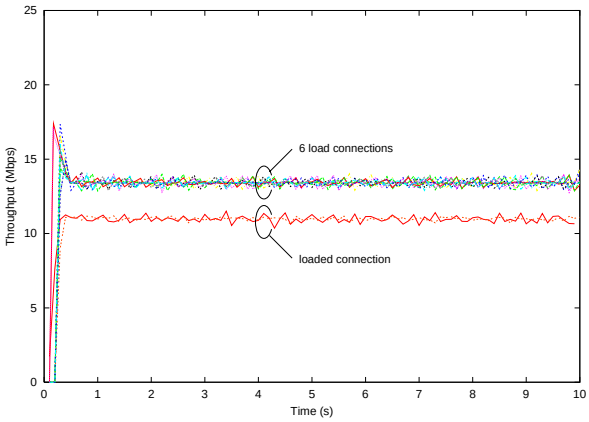


図 5: トラヒック測定実験 1 から得られたトラヒックのスループット推移

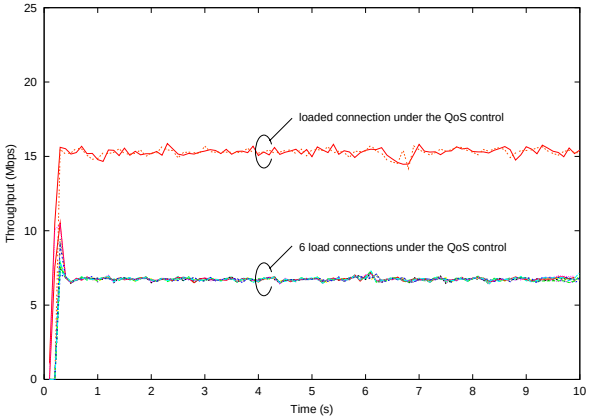


図 6: トラヒック測定実験 2 から得られたトラヒックのスループット推移

る．本実験では，ESPQ のトラヒックに対して WFQ による優先制御を行ったが，その他に TCP/UDP ヘッダの情報を用いた受付制御や帯域制御，輻輳制御を行うことができる．

# 5 諸考察

### ESPQ ヘッダの認証の安全性

ESPQ ヘッダの認証の安全性に関して考察する．ESPQ ヘッダは可変長の領域であり，8 バイトの UDP ヘッダが含まれたとき最も短くなる．即ち，ESPQ ヘッダ長は，16 バイト (UDP ヘッダ (8) + SPI(4) + Sequence Num(4)) になる．

この長さは，暗号化関数を用いたメッセージダイジェストの長さ，およびハッシュ関数のメッセージダイジェストの長さよりも同等あるいは，それ以上の長さである．これは，認証されるメッセージの情報量がメッセージダイジェストの情報量よりも等しいか，またはそれ以上であることを示している．

従って，ESPQ ヘッダの認証の安全性は，鍵付きハッシュ関数の安全性に基づいて保証される．

### ESPQ ペイロードデータの認証と暗号化の安全性

ESPQ ペイロードデータの認証および暗号化の安全性について考察する．ESPQ ペイロードデータは，(1) ハッシュ関数を用いて EPSQ ペイロードデータのメッセージダイジェストを計算し，(2)ESPQ ペイロードデータとそのメッセージダイジェストを暗号化する，ことで認証される．この認証方法は，ハッシュ関数の衝突困難性と，共通の暗号化鍵を持たない第三者が暗号文から平文を復元できないことを利用した認証方法である．

したがって，安全性が保証されているハッシュ関数と暗号化関数を用いるならば，ESPQ ペイロードデータの認証の安全性は保証される．ただし，同一の暗号文がネットワーク上に送信されないように，CBC モードや CFB モードの暗号化アルゴリズムを使用するべきである．一方，ESPQ ペイロードデータは，ESP と同様の方法で暗号化されている．したがって，ESPQ ペイロードデータの暗号化の安全性は ESP の暗号化の安全性と同等である．

### 不正なパケットの棄却能力

ESPQ の不正パケットの棄却能力を ESP の不正パケット棄却能力と比較しながら考察する．ESPQ は，(1)ESPQ ヘッダの認証，(2)ESPQ ペイロードデータの認証，という 2 つの認証を行う．(1) は，ESP のパケットの認証と同じ方法である．(2) は，復号とメッセージダイジェストの計算をともなうので，一見すると ESP に比べて不正パケットの棄却能力が劣っているように見えるが，次の理由から ESP の不正パケット棄却能力と本質的に変わらないことが言える．

- 攻撃者が (1) をパスする不正パケットを構築するには，通信を行っている正規のゲートウェイ (ホスト) 間に攻撃者が介在した person-in-the-middle 攻撃を行って，正規のパケットを横取りする必要がある．
- パケットの改竄を行う person-in-the-middle 攻撃は，正規のパケットが送信される時間間隔よりも短い間隔で不正パケットを構築できない．
- ESP と ESPQ は，無作為なパケットの偽造およびパケットの改竄を行う person-in-the-middle 攻撃から被る DoS 攻撃 および D-DoS 攻撃を防ぐことができない．

### TCP/UDP ヘッダの情報

ESPQ は，元パケットの TCP/UDP ヘッダの情報を秘匿しない．したがって，元パケットの TCP/UDP ヘッダの情報が安全な通信に及ぼす影響について考察する．

TCP/UDP ヘッダから得られる情報は次のとおりである．

- TCP ヘッダ
  送信元ポート番号，宛先ポート番号，シーケンス

番号，応答確認番号，コントロールフラグ，ウインドウサイズ，オプションペイロード

- UDP ヘッダ
  送信元ポート番号，宛先ポート番号

TCP ヘッダが含むシーケンス番号や応答確認番号，コントロールフラグは，従来，通信を行っているホストを攻撃する足がかりとして利用されてきたが，ESPQ は第三者によるパケットの改竄や偽造を防ぐので，通信を行う正規のホストがこれらの情報をもとに攻撃されることはない．また，第三者は TCP/UDP ヘッダのポート番号からアプリケーションの種類を特定したり，TCP ヘッダのウインドウサイズから通信帯域を把握できるが，それらの情報が ESPQ の安全な通信を脅かす直接の原因にはならない．しかし，ESPQ を利用する者は，少なくとも第三者に通信パケットの TCP/UDP ヘッダが閲覧されていることを意識するべきである．

## 6 まとめ

本稿では，バックボーン・ネットワーク上および組織内のプライベートネットワーク上で行われる QoS 制御を妨げないセキュリティ・プロトコル (ESP considered QoS; ESPQ) を提案した．

SSL や ESP は，QoS 制御に必要な元パケットの識別情報 (IP ヘッダの情報や TCP/UDP ヘッダの情報) を秘匿または隠蔽し，QoS 制御を妨げていた．ESPQ のパケットは，一般の TCP/UDP パケットと同様に，元パケットの識別情報が秘匿されることなくネットワーク上に公開され，ルータ等の制御機器によって QoS 制御が可能である．また，ESPQ のパケットは，元パケットの TCP/UDP ヘッダが IP ヘッダの直後に位置しているので，一般の TCP/UDP パケットと同様にネットワーク上で扱うことができる．

ESPQ は，QoS 制御を用いて安価に構築されたネットワーク環境において暗号化通信を行う際に有効なセキュリティ・プロトコルである．

## 参考文献


[1] S. Kent, R. Atkinson, “Security Architecture for the Internet Protocol,” Network Working Group, Request for Comments:2401, 1998.

[2] S. Kent, R. Atkinson, “IP Authentication Header,” Network Working Group, Request for Comments:2402, 1998.

[3] S. Kent, R. Atkinson, “IP Encapsulating Security Payload(ESP),” Network Working Group, Request for Comments:2406, 1998.

[4] D. Harkins, D. Carrel, “The Internet Key Exchange(IKE)”, Network Working Group, Request for Comments:2409, 1998.

[5] C. Madson, R. Glenn, “The Use of HMAC-MD5-96 within ESP and AH,” Network Working Group, Request for Comments:2403, 1998.

[6] C. Madson, R. Glenn, “The Use of HMAC-SHA-1-96 within ESP and AH,” Network Working Group, Request for Comments:2404, 1998.

[7] H. Krawczyk, M. Bellare, R. Canetti, “HMAC: Keyed-Hashing for Message Authentication,” Network Working Group, Request for Comments:2104, 1997.

[8] C. Madson, N. Doraswamy, “The ESP DES-CBC Cipher Algorithm With Explicit IV,” Network Working Group, Request for Comments:2405, 1998.

[9] R. Pereira, R. Adams, “ESP CBC-Mode Cipher Algorithms,” Network Working Group, Request for Comments:2451, 1998.

[10] M. Bellare, R. Canetti, H. Krawczyk, “Keyed Hash Functions and Message Authentication,” Proceedings of Crypto’96, LNCS 1109, pp.1–15, 1996.

[11] R. Rivest, “The MD5 Message-Digest Algorithm,” Network Working Group, Request for Comments:1321, 1992.

[12] FIPS 180–1, *Secure Hash Standard*, NIST, U.S. Department of Commerce, Washington D.C., 1995.

[13] W. Tuchman, “Hellman Presents No Shortcut Solutions to DES,” IEEE Spectrum, vol.16, no.7, pp.40–41, 1979.

[14] R. Baldwin, R. Rivest, “The RC5, RC5-CBC, RC5-CBC-Pad, and RC5-CTS Algorithms,” Network Working Group, Request for Comments:2040, 1996.

[15] 岡本, 山本: 現代暗号, 産業図書, 1997.

[16] P. Ferguson, G. Huston: Quality of Service:Delivering QoS on the Internet and in Corporate Networks, John Wiley & Sons,Inc., 1998.

[17] B. Schneier: Applied Cryptography, 2nd Edition, John Wiley & Sons,Inc., 1996.

[18] W. Stallings: CRYPTOGRAPHY AND NETWORK SECURITY:Principles and Practice, Prentice Hall, 1998.

[19] DBS：A TCP Benchmark Tool, *available at http://shika.aist-nara.ac.jp/member/yukio-m/dbs/*

[20] SSL 3.0 SPECIFICATION,
*available at http://home.netscape.com/eng/ssl3/*