2004 年度 修士論文

ポート制御を可能にする IPsec 仕様の拡張法

提出日： 2005 年 2 月 2 日

指導：後藤滋樹教授

早稲田大学 大学院理工学研究科情報・ネットワーク専攻
学籍番号： 3603U031-8

岡部吉彦

目次

図一覧

表一覧

第 1 章

序論

情報通信技術やインターネットの整備・拡大により、情報流通の高速化・ボーダレスが飛躍的に進展しつつある。インターネットを介して相互接続されるコンピュータの台数は年々増加しており、物理的な距離を考慮することなく手軽に遠方のコンピュータへのアクセスが可能となった。今や、インターネットは現代社会において欠くことのできない存在となっている。最近では、外出先などからインターネットを使って安全に社内へアクセスしたり、特定のビジネスパートナーに対して安全に情報提供したりするニーズが高まっている。このようなニーズに対して専用線を用いる方法があるが、コストがかかってしまう問題があった。インターネットを利用した場合にはコストの削減が可能であるが、データの盗聴・改ざんの危険が存在する。この両方の問題を改善するものとして VPN (Virtual Private Network) が考えられた。 VPN は、インターネット上の拠点間を専用線のように接続し、盗聴や改ざんなどの不正行為を防ぎ、安全な通信を可能にする技術である。インターネットを経由しているにもかかわらず、あたかも同一のネットワーク上にいるかのような利便性が得られる。

VPN に使われる技術の 1 つに IPsec がある。本論文では、この IPsec について、アプリケーションごとに制御できるように機能の追加を行う。

第 2 章

IPsec と VPN

2.1 VPN の機能

VPN には、理想的に専用線と同等の性質が求められる。しかし、実際には専用線と全く同じ性質を実現するの困難であり、現在の VPN では、次に挙げる機能のうち一部を実現している。

トンネリング機能

専用線では、データの種類や内容に関係なく、一方の接続点から入力されたデータは、もう一方の接続点へと出力される。 VPN では、この機能を実現するために、トンネリングという技術を使用する。トンネリングによって、ネットワーク上の 2 地点間を結ぶ仮想的な通進路を構築することができる。

データのセキュリティの確保

専用線を使用した場合には、第三者にデータを盗聴される危険性はあまりないが、インターネットを使用した場合は、データを盗聴される危険性がある。このため、 VPN ででは、データの内容が第三者に知られないように機密性を確保する必要がある。また、データの改ざんやなりすましから保護する必要もある。

マルチプロトコル転送

専用線では、 IP を含むさまざまなネットワーク層プロトコルを使用することが可能である。組織内で、 IP 以外のプロトコルを使用している場合には、 VPN がマルチプロトコル転送に対応している必要がある。

シーケンス保証

専用線ではフレームの到着順序が保証されるが、インターネットのようなパケット交換網では到着順序が保証されない。そこで、パケットに付与されたシーケンス番号をチェックすることにより、送信順序を確認する機能をもつ VPN も存在する。

サービス品質 (QoS : Quality of Service) の保証

専用線では、帯域や遅延の割合などのサービス品質 (QoS) が保証される。 VPN においても、このような QoS を保証する仕組みが求められる。 QoS を確保するための技術としては、 RSVP (Resource Reservation Protocol) や DiffServ (Differentiated Services) などが提案されており、VPN で利用することができる。

2.2 VPN の形態

VPN の形態には、大きく拠点間接続 VPN とリモートアクセス VPN の 2 つがある。

拠点間接続 VPN

離れた拠点同士を VPN で接続する形態。

リモートアクセス VPN

プロバイダのアクセスポイントなどにダイヤルアップ接続した端末と企業ネットワークとの間で VPN を構築する形態。

2.3 トンネリングプロトコル

トンネリングプロトコルとは、 VPN を構築するために必要なトンネリング機能を持つプロトコルのことである。 IPsec もこのトンネリングプロトコルの 1 つである。ここでは VPN を実現する上で多く利用されている L2TP (Layer 2 Tunneling Protocol) と MPLS (Multiprotocol Label Switching) を紹介する。 IPsec については、次節で説明する。

2.3.1 L2TP

L2TP は、 Microsoft 社などが開発した PPTP (Point-to-Point Tunneling Protocol) と Cisco Systems 社が開発した L2F (Layer 2 Forwarding) を基に標準化されているトンネリングプロトコルである。 L2TP は、データリンク層プロトコルである PPP (Point-to-Point Protocol) フレー

ムをカプセル化する。ただし、 L2TP には、セキュリティを確保する機能がないので、 IPsec と併用することで、セキュリティを確保する。しかし、この場合には追加するヘッダの量が多くなり、転送時のオーバーヘッドが大きくなるという欠点がある。

2.3.2 MPLS

MPLS は、 IETF で標準化されているラベルスイッチング技術である。 MPLS を使用した VPN では、 IP ヘッダの前に 4 バイトのシムヘッダ (Shim Header) を挿入し、このシムヘッダ中のラベルを参照することにより、 IP パケットを高速で転送する。 MPLS による配送の間は、 IP アドレスは参照されないため、プライベートアドレスを使用することが可能である。ただし、 MPLS を使用するためには、 MPLS に対応したルータでネットワークを構築する必要がある。

2.4 IPsec

IPsec は、セキュリティ機能のない IPv4 と、次世代の IP である IPv6 の両方のセキュリティを確保するプロトコルである。特に、 IPv6 では、 IPsec の実装が必須となっており、標準で使用することができる。

2.4.1 IPsec プロトコル

IPsec は、複数のプロトコルから構成されるプロトコルの集合 (プロトコルスイート) である。表 2.1 に、 IPsec で使用されるプロトコルをまとめる。

表 2.1: IPsec で使用されるプロトコル

プロトコル名称	プロトコル番号 など
認証ヘッダ (AH : Authentication Header)	プロトコル番号 51
暗号ペイロード (ESP : Encapsulation Security Payload)	プロトコル番号 50
IPComp (IP Payload Compression Protocol)	プロトコル番号 108
IKE (Internet Key Exchange)	500/UDP

IPComp と IKE は、 IPsec とは独立したプロトコルであるが、 IPsec と組み合わせて使用されるため、 AH、 ESP、 IPComp、 IKE の 4 つのプロトコルをまとめて IPsec と呼ぶ場合がある。各プロトコルについて次章で説明していく。

2.4.2 トンネルモードとトランスポートモード

IPsec では、カプセル化モードとして、トランスポートモードとトンネルモードがある。セキュリティゲートウェイでは、トンネルモードを実装する必要がある。 IPsec ホストでは、トンネルモードとトランスポートモードの両方をサポートする必要がある。

トランスポートモード

ホスト間で IPsec を使用する場合には、トランスポートモードを使用する (図 2.1)。トランスポートモードでは、主に IP ペイロード部のセキュリティを確保する。

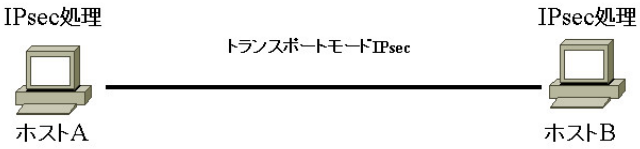

図 2.1: ホスト間におけるトランスポートモード IPsec

IPv4 では、 IPv4 ヘッダとトランスポート層プロトコルヘッダの間に IPsec で使用するヘッダを挿入する (図 2.2)。網掛部は、暗号化部分である。

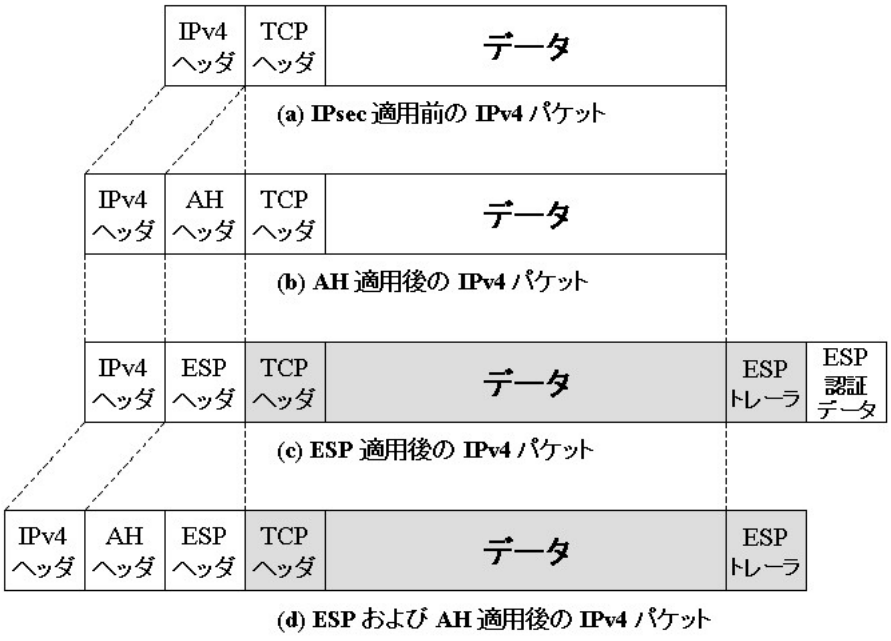

図 2.2: トランスポートモード IPsec を適用した IPv4 パケット

IPv6 では、 AH ヘッダおよび ESP ヘッダは IPv6 拡張ヘッダとして定義されている。 IPv6 の場合も IPv4 の場合と同様に、 IPv6 ヘッダとトランスポート層プロトコルヘッダの間に IPsec で使用するヘッダを挿入する (図 2.3)。網掛部は、暗号化部分である。

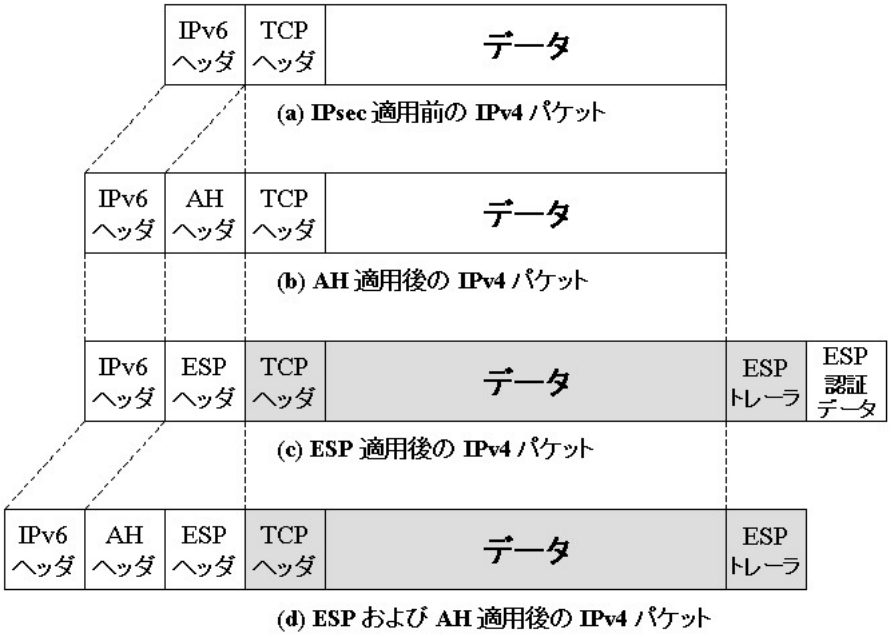

図 2.3: トランスポートモード IPsec を適用した IPv6 パケット

トンネルモード

トンネルモードは、主にセキュリティゲートウェイ間で IPsec トンネルを構築する場合に使用される。トンネルモードでは、 IP パケット全体に対してセキュリティ機能を適用する。そして、セキュリティゲートウェイが新たにトンネル配送用の IP ヘッダを追加して、相手側のセキュリティゲートウェイに送信する。こうすることで VPN を構築することが可能になる。

セキュリティゲートウェイ間 (図 2.4) でトンネルモードを使用する場合は、パケットを送信するホストで IPsec を実装する必要はなく、トンネルの始点と終点にとなるセキュリティゲートウェイが IPsec の機能を実現する。

また、セキュリティゲートウェイ間だけでなく、セキュリティゲートウェイとホストの間 (図 2.5)、ホスト間 (図 2.6) でトンネルモードを利用することも可能である。

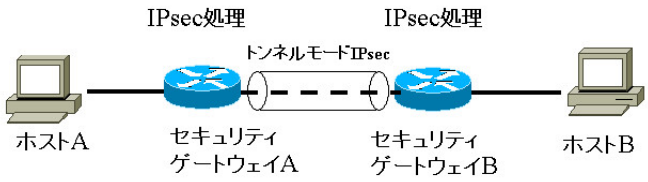

図 2.4: トンネルモード IPsec (セキュリティゲートウェイ間)

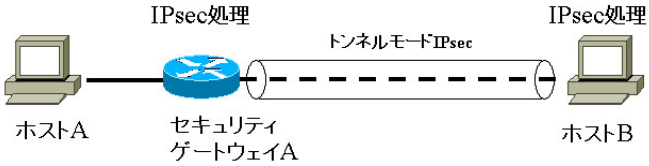

図 2.5: トンネルモード IPsec (セキュリティゲートウェイとホスト間)

図 2.6: トンネルモード IPsec (ホスト間)

IPv4 パケットに対してトンネルモード IPsec を適用する場合は、オリジナルの IPv4 パケット全体に対して IPsec のセキュリティ機能を適用し、そのパケットをトンネルの終点まで運ぶための IPv4 ヘッダと IPsec で使用するヘッダを新たに追加する。図 2.7 は、その構成を表している。なお、網掛部は暗号化部分である。

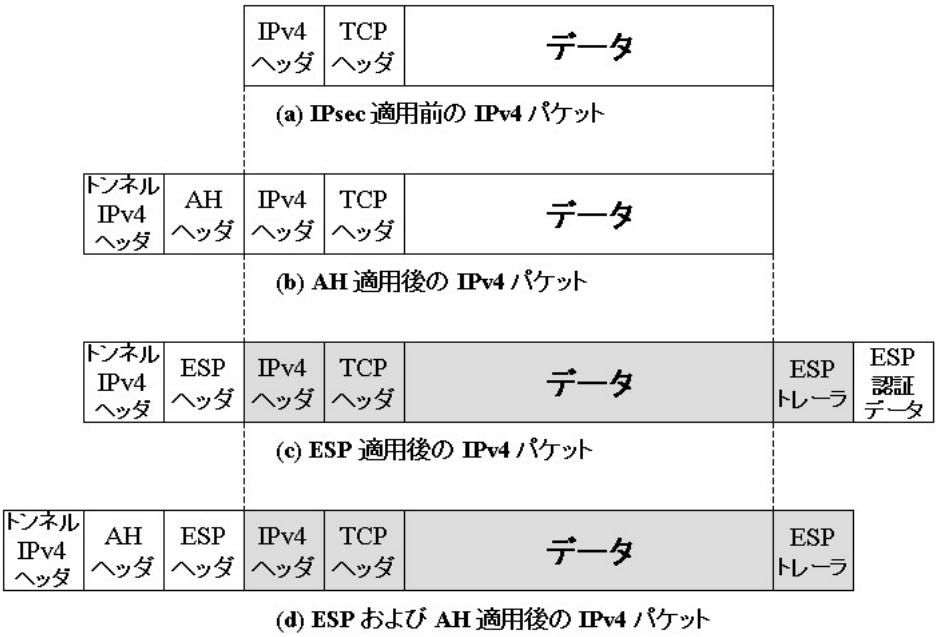

図 2.7: トンネルモード IPsec (IPv4) を適用した IPv4 パケット

IPv6 パケットに対してトンネルモード IPsec を適用する場合は、オリジナルの IPv6 パケット全体に対して IPsec のセキュリティ機能を適用し、そのパケットをトンネルの終点まで運ぶための IPv6 ヘッダと IPsec で使用するヘッダを新たに追加する。図 2.8 は、その構成を表している。なお、網掛部は暗号化部分である。

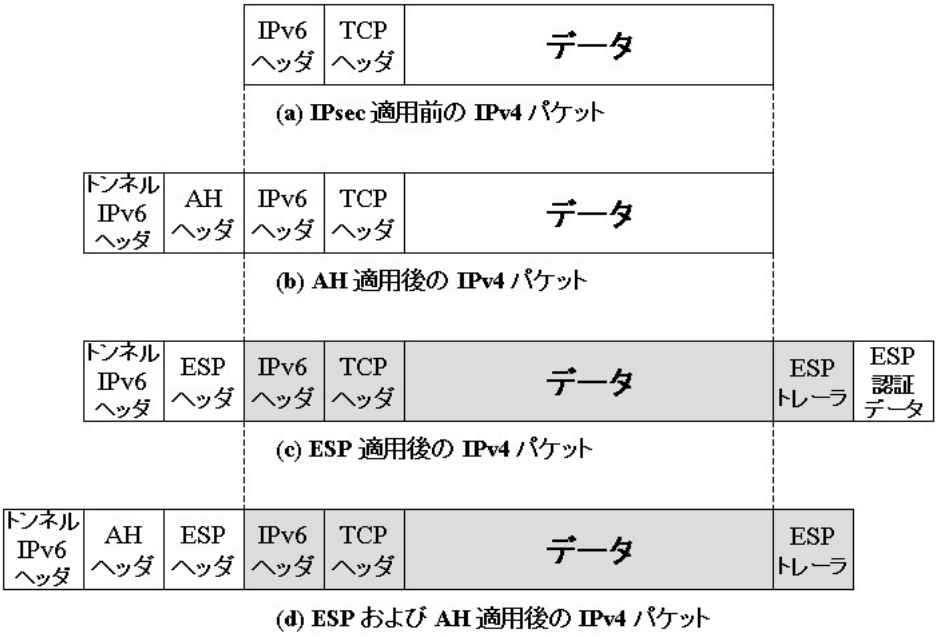

図 2.8: トンネルモード IPsec (IPv6) を適用した IPv6 パケット

上記の他に、 IPv6 パケットに対して IPv4 トンネルモード IPsec を適用し、 IPv4 ネットワークを配送させることや、 IPv4 パケットにに対して Pv6 トンネルモード IPsec を適用し、 IPv6 ネットワーク上を配送させることも可能である。

2.4.3 セキュリティポリシー

セキュリティポリシーによるパケットの処理

IPsec を実装したノードでは、 IP パケットの処理方法を記述したセキュリティポリシーが必要となる。 IPsec におけるセキュリティポリシーでは、 IP パケットに対して、次のうちどの処理を行うかを記述する。

- パケットを破棄する (discard)
- IPsec を適用せずに通常の処理を行う (bypass IPsec)
- IPsec を適用する (apply IPsec)

セレクタ

セキュリティポリシーにおいて処理対象となるパケットを指定するために利用する情報をセレクタと呼ぶ。セレクタには、パケットの始点アドレス、終点アドレス、トランスポート層プロトコル、送信元ポート番号、宛先ポート番号などが使用される。

セキュリティポリシーデータベース (SPD)

IPsec のセキュリティポリシーは、セキュリティポリシーデータベースに登録される。 SPD には、出力パケットに対するセキュリティポリシーが格納された出力用 SPD と、入力パケットに対するセキュリティポリシーが格納された入力用 SPD の 2 種類が存在する。セキュリティポリシーの適用対象となるパケットは、セレクタによって指定する。

2.4.4 セキュリティアソシエーション

IPsec では、セキュリティアソシエーション (SA : Security Association) という概念を導入している。 SA は、あるトラフィックのセキュリティを確保する単方向のコネクションである。

SA におけるセキュリティは、 AH や ESP などのセキュリティプロトコルよって確保される。 SA はセキュリティプロトコルごとに確立されるため、 AH と ESP によって確立される SA は別々のものである。また、単方向のコネクションであるため、実際の通信には、方向の異なる 2 本の SA が使用される。図 2.9 は、 AH と ESP を使用したときの SA の確立状況である。

SA を識別するための情報として、終点アドレス、 AH や ESP などのセキュリティプロトコルの種別、セキュリティパラメータインデックス (SPI : Security Parameter Index) の 3 つが使用される。つまり、終点とセキュリティプロトコルが異なると、別の SA が使用される。 SPI は、終点とセキュリティプロトコルが同じ SA を多重化するための識別子として使用される。

SA では、カプセル化モード (トランスポートモードまたはトンネルモード)、使用する IPsec プロトコルの種別 (AH または ESP)、 AH や ESP で使用するアルゴリズム (暗号化アルゴリズムや認証アルゴリズム) などが決められている。これらの情報は、手動で設定することもできるが、 IKE などの SA を管理するプロトコルによって自動的にで折衝することも可能である。

図 2.9: ホスト間に確立される SA(AH と ESP を使用した場合)

SA に関する情報は、セキュリティアソシエーションデータベース (SAD : Security Associa-

tion Database) に格納される。 SAD は手動で管理することも可能であるが、 IKE を使用することによって自動的に管理することができる。 SAD には、出力パケットの処理の際に参照される出力用 SAD と、入力パケットの処理の際に参照される入力用 SAD がある。 SAD の各エントリには、キーとなる終点アドレス、 IPsec プロトコルの種別、 SPI 値の他に、 SA に関する情報が記述される。

2.4.5 IPsec 処理の流れ

出力処理

IP パケットが IPsec 機器から出力される場合には、以下のようにして処理を行う。

1. セキュリティポリシーの検索

   対象となる IP パケットの始点アドレス、終点アドレス、上位層プロトコル、ポート番号などのセレクタをキーとして、出力用 SPD からセキュリティポリシーを検索する。

2. SA の検索

   検索されたセキュリティポリシーが、「IPsec を適用 (apply IPsec)」である場合には、出力用 SAD から該当する SA の情報を検索する。該当する SA が存在しない場合は、新しい SA を生成する。該当する SA が存在した場合は、その SA に関する情報を取り出す。

3. IPsec プロトコルの処理

   該当する IPsec プロトコル (AH または ESP) において処理をする。

4. パケット送信

   IPsec 処理が適用されたパケットを送信する。

入力処理

IPsec が適用されているパケットが入力された場合には、以下のようにして処理を行う。

1. SA の検索

   SPI 値、終点アドレス、 IPsec プロトコル (AH または ESP) をキーとして、入力用 SAD から該当する SA の情報を検索する。該当する SA が存在しなかった場合には、 IPsec パケットを廃棄する。

2. IPsec プロトコルの処理

   該当する SA が存在した場合には、アルゴリズムや鍵を取り出して、 IPsec プロトコル (AH または ESP) の処理を行う。

3. セキュリティポリシーの検索

   AH や ESP の処理が行われた後の IP パケットの始点アドレス、終点アドレス、上位層プロトコル、ポート番号などのセレクタをキーとして入力用 SPD から該当するセキュリティポリシーを検索する。そして、検索されたセキュリティポリシーの内容と適用されていた IPsec 処理の内容が合致していることを確認する。合致しない場合には、パケットを破棄する。

4. その後の処理

   自身宛の IP パケットである場合には、データをトランスポート層に渡す。別のホスト宛の IP パケットである場合には、そのホストに IP パケットを転送する。

第 3 章

セキュリティプロトコルの詳細

3.1 AH

3.1.1 AH の機能

コネクションレス完全性の確保

AH ヘッダに含まれている完全性チェック値 (ICV: Integrity Check Value) を検証することで IP ヘッダの一部と IP ペイロード部全体の完全性を確保することが可能となる。ここで、 IP ヘッダの一部というのは、配送中に中継ルータなどによって変更されない IP ヘッダのフィールドを指す。具体的には、 TTL (IPv6 では中継限界数) や TOS (IPv6 ではトラフィッククラスおよびフローラベル)、チェックサム等である。これらのフィールドは、 AH で保護することはできない。

データ送信元の認証

AH ヘッダに含まれている ICV を検証すると、その IP パケットが、事前に安全な方法で共有された共有秘密鍵を持つ相手から送信されたものであることを確認することができる。

リプレイ防御

AH ヘッダに含まれているシーケンス番号をチェックすることで、再送されたパケットの受信を拒否することが可能になる。ただし、この機能はオプションである。

3.1.2 AH プロトコルフォーマット

AH のヘッダフォーマットは図 3.1 のようになっている。

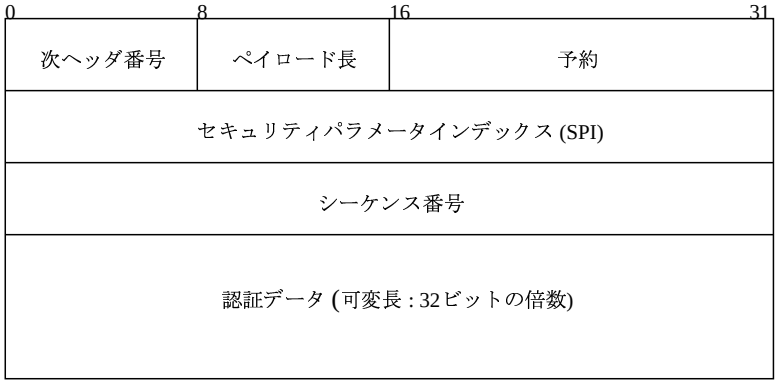

図 3.1: AH のヘッダフォーマット

次ヘッダ番号フィールド 次ヘッダ番号フィールドは、 8 ビットの固定フィールドである。このフィールドでは、 AH の次に挿入されるヘッダの IP プロトコル番号が入る。

ペイロード長フィールド ペイロード長フィールドは、 8 ビットの固定長フィールドである。このフィールドには、 32 ビットワード単位での AH ヘッダの長さから 2 を引いた値が入る。

予約フィールド 予約フィールドは、 16 ビットの固定長フィールドである。このフィールドは、将来の利用のために予約されており、すべてのビットが 0 にセットされている必要がある。

セキュリティパラメータインデックス (SPI) フィールド セキュリティパラメータインデックスフィールドは、 32 ビットの固定長フィールドである。このフィールドには、 SA を識別するための SPI 値が入る。

シーケンス番号フィールド シーケンス番号フィールドは、 32 ビットの固定長フィールドである。このフィールドには、パケットを送信する際に 1 ずつ増加するシーケンス番号が入る。このシーケンス番号は、 SA が生成された際に 0 に初期化されるため、 SA 生成後の最初のパケットに含まれるシーケンス番号は 1 となる。

認証データフィールド 認証データフィールドには、このパケットに対する完全性チェック値 (ICV) が入る。このフィールドは可変長だが、 AH ヘッダ全体の長さは、 IPv4 の場合は 32 ビットの整数倍、 IPv6 の場合は、 64 ビットの整数倍である必要があるため、 ICV の後に必要に応じてパディングを挿入して AH ヘッダ全体の長さを調整する。

3.1.3 認証アルゴリズム

AH では、認証アルゴリズムとして HMAC-MD5-96、 HMAC-SHA-1-96 の実装が必須とされている。この他の認証アルゴリズムには、 HMAC-RIPEMD-160-96、 DES-MAC、 AES-MAC、

Keyed-MD5 がある。

3.2 ESP

3.2.1 ESP の機能

データの機密性確保

ESP のメインとなる機能がデータの暗号化機能である。トンネルモードの場合は IP パケット全体を暗号化し、トランスポートモードの場合は IP ペイロード部暗号化することにより、データの機密性を確保する。

コネクションレス完全性の確保

ESP で認証機能を有効にした場合に含まれる完全性チェック値を検証することで、 IP ペイロード部の完全性を確保することが可能となる。ただし、 AH と異なり、パケット配送用の IP ヘッダの完全性を確保することはできない。 ESP では、この機能はオプションとして用意されている。

データ送信元の認証

ESP で認証機能を有効にした場合に含まれる ICV を検証することで、その IP パケットが、共有秘密鍵を持つ相手から送信されたものであることを確認することが可能になる。 ESP では、この機能はオプションとして用意されている。

リプレイ防御

認証機能を有効にしている場合は、リプレイ防御機能を有効にすることが可能である。 ESP ヘッダに含まれるシーケンス番号をチェックすることで、過去に受信した IP パケットを再び受信することを防ぐことが可能である。 ESP では、この機能はオプションとして用意されている。

トラフィック情報の機密性確保

トンネルモードを使用して IP パケット全体を暗号化することにより、データの送信元、宛先、利用プロトコルなどの情報を隠蔽することが可能である。また、暗号化の前にパディングをある程度付加して送信量を多くすることで、実際のデータ量をある程度隠蔽することがも可能である。

3.2.2 ESP プロトコルフォーマット

ESP のパケットフォーマットは図 3.2 のようになっている。

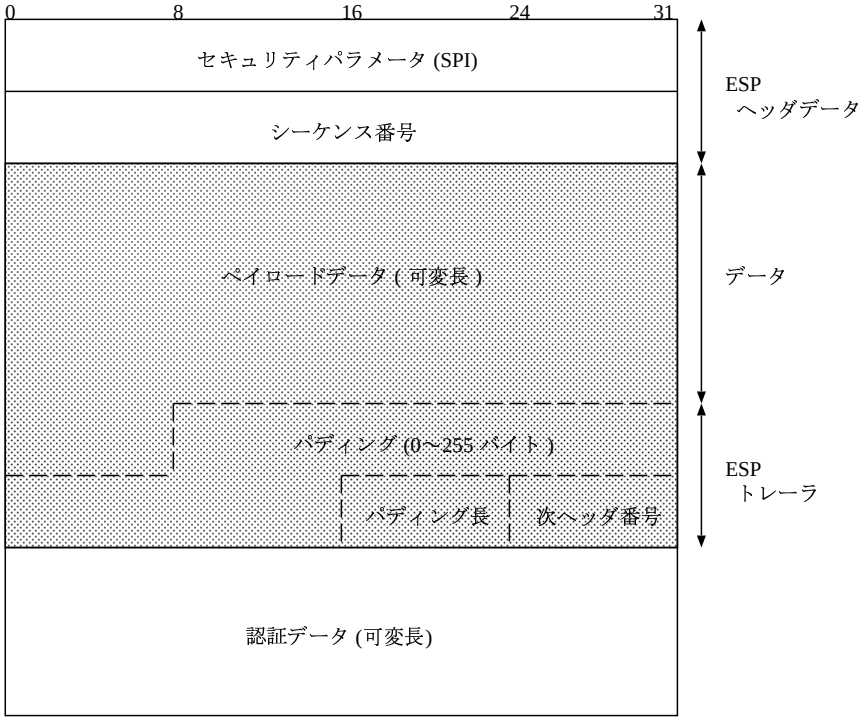

図 3.2: ESP のパケットフォーマット (網掛部は暗号化)

セキュリティパラメータインデックス (SPI) フィールド セキュリティパラメータインデックスフィールドは、32 ビットの固定長フィールドである。このフィールドには、SA を識別するための SPI 値が入る。

シーケンス番号フィールド シーケンス番号フィールドは、32 ビットの固定長フィールドである。このフィールドには、パケット送信する際に 1 ずつ増加するシーケンス番号が入る。このシーケンス番号は、SA が生成された際に 0 に初期化されるため、SA 生成後の最初のパケットに含まれるシーケンス番号は 1 となる。

ペイロードデータフィールド (暗号化) ペイロードデータフィールドは、整数バイト長の可変長フィールドである。トランスポートモードの場合は、このペイロードデータフィールドには暗号化された IP ペイロード部が入る。また、トンネルモードの場合には、暗号化された IP パケット全体が入る。暗号化アルゴリズムで初期ベクトル (IV) を必要とする場合は、このペイロードデータフィールド中で暗号化されていない状態で運ばれる。

パディングフィールド (暗号化) ペイロードデータフィールドとパディング長フィールドの間にパディングフィールドが挿入される。このパディングの目的は、次の通りである。

- パディング長フィールドと次ヘッダ番号フィールドが 4 バイト境界の右端に位置するように調整する
- IV を除いたペイロードデータフィールドとパディングフィールド、パディング長フィールド、次ヘッダ番号フィールドの合計の長さが、使用する暗号化アルゴリズムのブロック長の整数倍となるように調整する
- トラフィック情報の機密性を確保するため、送信するデータの量を実験のデータ量よりも多くする

パディングの値が暗号化アルゴリズムで指定されない場合は、デフォルトで 1 バイトごとに 1、 2、 3・・・と増加する整数値 (最初のパディングの値は 1、 2 つめのパディングの値は 2) が入る。

パディング長フィールド (暗号化) パディング長フィールドは、 8 ビットの固定長フィールドである。このフィールドには、直前のパディングの長さがバイト単位で入る。

次ヘッダ番号フィールド 次ヘッダ番号フィールドは、 8 ビットの固定長フィールドである。このフィールドには、ペイロードデータフィールドを復号化した際に最初に現れるヘッダの IP プロトコル番号が入る。

認証データフィールド 認証データフィールドは、可変長フィールドである。このフィールドは、認証機能を有効にした場合のみ存在する。このフィールドには、 ESP ヘッダ、暗号化されたデータ、 ESP トレーラに対する完全性チェック値 (ICV) が入る。認証データフィールドの長さは、 SA で指定された認証アルゴリズムの出力の長さによって決定される。

3.2.3 暗号化アルゴリズム

ESP では、暗号化アルゴリズムとして DES-CBC と NULL 暗号化アルゴリズムの実装が必須となっている。その他に使用可能な暗号化アルゴリズムは、 3DES-CBC、 RC5-CBC、 IDEA-CBC、 CAST-128-CBC、 Blowfish-CBC がある。 NULL 暗号化アルゴリズムは、実際には暗号化を行わないため、必ず認証アルゴリズムを指定する必要がある。

3.2.4 認証アルゴリズム

ESP で認証機能を有効にした場合には、 AH と同様の認証アルゴリズムが使用される。

3.3 IPComp

IPComp は、 IP のレベルでデータを圧縮するプロトコルである。 IPComp の仕様は、 RFC2393 に記述されている。

3.3.1 IP レベルでの圧縮の必要性

データを圧縮することによって、データ転送のスループットを高めることができる。ただし、どのような場合でも圧縮が有効に機能するとは限らない。圧縮は、データの規則的な並びを、ある法則にしたがって短い情報で表すことによってデータ量を削減している。つまり、データに規則的な並びが多いほど、圧縮率が高くなるが、規則的な並びが少ないと圧縮率が悪くなる。場合によってはデータ量が増えてしまうことがある。特に、暗号化されたデータには規則的な並びがほとんど発生しない。そこで、 ESP によってデータが暗号化される前に圧縮を行う必要がある。これを実現するプロトコルが IPComp である。 IPComp は ESP による暗号化処理の前にデータを圧縮するため、効率よく圧縮を行うことが可能である。

3.3.2 IPComp アソシエーション

IPComp を使用するためには、事前に送信側と受信側との間で IPComp アソシエーション (IPCA: IPComp Association) を確立する必要がある。 IPCA は、 IPsec のセキュリティアソシエーション (SA) と似ており、次の内容が定義されている。

- 複数の IPCA を識別するための CPI (Compression Parameter Index: 圧縮パラメータ)
- 使用する圧縮アルゴリズム
- カプセル化モード (トンネルモードまたはトランスポートモード)
- IPCA の有効期間

IPCA のパラメータは手動で設定することも、 IKE によって自動的に折衝することも可能である。

3.3.3 IPComp プロトコルフォーマット

IPComp のヘッダフォーマットは図 3.3 のようになっている。

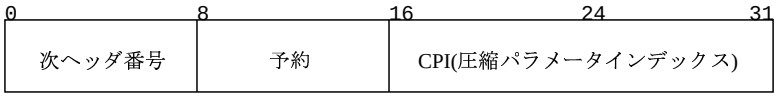

図 3.3: IPComp ヘッダの構成

次ヘッダ番号フィールド

次ヘッダ番号フィールドは、8 ビットの固定長フィールドである。このフィールドには、圧縮されたデータにおける最初のプロトコルヘッダの番号が入る。

予約フィールド

予約フィールドは、8 ビットの固定長フィールドである。このフィールドは、将来のために予約されており、すべてのビットは 0 にセットされている必要がある。

CPI フィールド

CPI フィールドは、 CPI 値が入る 16 ビットの固定長フィールドである。

3.3.4 圧縮アルゴリズム

IPComp で使用可能な圧縮アルゴリズムとしては、 DEFLATE と LZS、 LZJH の 3 つがある。IPComp は特定のアルゴリズムに依存しないように設計されているため、この他の圧縮アルゴリズムが将来追加される可能性もある。

3.3.5 IPComp 処理の流れ

まず、送信側での処理を説明する。トンネルモードの場合は、 IP パケット全体が圧縮対象になり、トンラスポートモードの場合は、トランスポート層ヘッダ以降の部分が圧縮対象になる。圧縮処理後、圧縮したデータの前に IPComp ヘッダを追加する。その後、 ESP や AH の処理を行い、受信側に送信する。

受信側では、 ESP や AH の処理を行い、 IPComp ヘッダの CPI 値から使用されている圧縮アルゴリズムを判断し、データの復元を行う。

図 3.4 (トランスポートモード)、図 3.5 (トンネルモード) は、 IPComp 処理を行った場合の IPsec (ESP) のパケット構成を表している。斜線部は圧縮部分、網掛部は暗号化部分を表している。

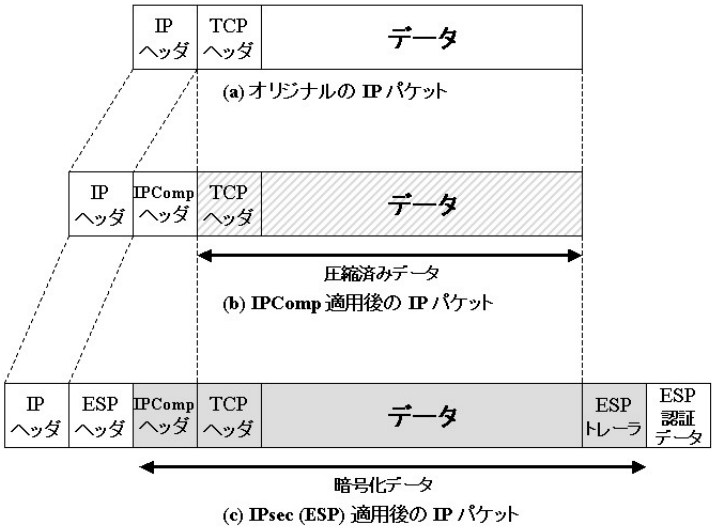

図 3.4: IPComp を行った場合のパケット構成 (トランスポートモード)

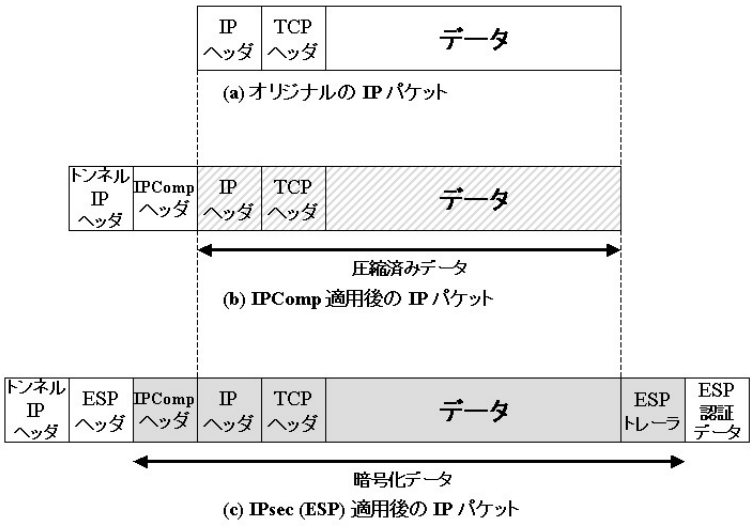

図 3.5: IPComp を行った場合のパケット構成 (トンネルモード)

3.4 IKE

3.4.1 IPsec における鍵管理プロトコルの構成

IPsec では、 SA と共有秘密鍵を管理するためのフレームワークとして ISAKMP (Internet Security Association and Key Management Protocol) を定めている。さらに、 ISAKAMP フレームワーク上で動作する鍵管理プロトコルとして IKE (Internet Key Exchange) を定めている。ただし、 IKE は、 IPsec での使用に特化したものではなく、汎用的に利用できるプロトコルとして開発されたものである。

ISAKMP

ISAKMP は、セキュリティプロトコルの SA と鍵の管理を行うためのプロトコルのフレームワークである。このフレームワークでは、プロトコルのフォーマットやメッセージ交換の手順について定めているが、実際の鍵交換の仕組みについて定めていない。

IKE

実際の鍵交換の方法を定めたものが、 IKE である。

ISAKMP SA

IPsec SA で使用されるパラメータの折衝を保護するために、 ISAKMP セキュリティアソシエーション (ISAKMP SA) が使用される。 ISAKMP SA は、セキュリティプロトコルの SA を折衝する際に使用されるメッセージの機密性と完全性の確保や、通信相手の認証、リプレイ防御などのセキュリティ機能を提供する。

3.4.2 IKE の持つ機能

秘密鍵を共有する際には、相手を認証する必要がある。 IKE では、 AH または ESP で使用する秘密鍵を共有する前に相手を認証する。 IKE では、次の 4 種類の相手認証方法をサポートしている。

1. 事前共有秘密鍵認証方法
2. デジタル署名認証方式 (RSA、 DSA、 ECDSA)
3. 公開鍵暗号化認証方式 (RSA、 ElGamal)
4. 改良型公開鍵暗号化認証方式 (RSA、 ElGmal)

3.4.3 IKE の動作

IKE は 2 つのフェーズから成り立っている。フェーズ 1 では ISAKMP SA を確立し、フェーズ 2 では IPsec などのセキュリティプロトコルの SA を確立する。フェーズ 2 では、フェーズ 1 で折衝された ISAKMP SA によってセキュリティが確保される。

ISAKMP SA の確立 (フェーズ)

フェーズ 1 では次のことが行われる。

1. ISAKMP SA の折衝
2. ISAKMP SA で使用される共有秘密鍵
3. 相手の認証

セキュリティプロトコルの SA の確立 (フェーズ 2)

フェーズ 2 では、フェーズ 1 で確立された ISAKMP SA を使用して、 IPsec の AH や ESP などのセキュリティプロトコルの SA を確立する。フェーズ 2 では次のことが行われる。

1. セキュリティプロトコルの SA の折衝
2. セキュリティプロトコルの SA で使用される共有秘密鍵の生成

第 4 章

IPsec の問題点と解決法の提案

4.1 IPsec の問題点

4.1.1 ファイアフォールの制御に関する問題

ファイアフォール装置の背後に、IPsec 装置を設置した場合には、IPsec の通信を許可するために、下記を許可する必要がある。

- プロトコル番号 51 (AH)
- プロトコル番号 50 (ESP)
- プロトコル番号 108 (IPComp)
- UDP ポート 500 (IKE)

ただし、上記のものを許可した場合には、IPsec を使ったすべての通信を許可することになる。IPsec の通信がすべて許可して良いものであれば問題ないが、以下の場合はどうだろうか。

- ファイアフォールの内側にセキュリティゲートウェイがあり、本来なら許可していないサービスを利用している。
- あるホストがコンピューターウィルスに感染していて、外部へ特定のポートを使用して通信している。

このような場合には、ポートごとに制御が出来る方が望ましいと考えられる。

4.2 解決法の提案

IPsec を行う通信が、 TCP または UDP の場合には、そのトランスポート層のヘッダを、新たに作られる IP ヘッダと ESP ヘッダまたは AH ヘッダの間に挿入にする。図 4.1 はトンネルモード、図 4.2 はトランスポートモードにおける ESP パケットの構成である。

図 4.1: ESP の場合のパケット構成 (トンネルモード)

図 4.2: ESP の場合のパケット構成 (トランスポートモード)

ただし、単にトランスポートヘッダを追加しただけでは、 IPsec 用に追加したトランスポートヘッダなのか、オリジナルのトランスポートヘッダなのかの区別がつかない。そのため、 IP ヘッダのオプション部分にすべて 0 の 4 バイトのフィールドを識別情報として使用する (図 4.3)。

図 4.3: IP オプションの使用

仮に、トランスポートヘッダが、 TCP ヘッダ (図 4.4) である場合には、宛先ポート番号、送信元ポート番号、制御フラグのみ複製し、チェックサムフィールドを除いた残りのフィールドは、すべて 0 とする。チェックサムフィールドは、改めて計算した値を使用する。これは、必要最低限の情報だけを残すことにより、盗聴された際にできるだけ情報を与えないためである。

図 4.4: TCP ヘッダ

第 5 章

実験

前章での構成を実装して、実際にフィルタリングが可能かどうか確かめてみる。また、疑似ヘッダを使用しない場合 (従来方法) と送信時間の比較をして、遅延時間を測定する。

5.1 実験環境

図 5.1 のようなマシン構成で実験を行った。

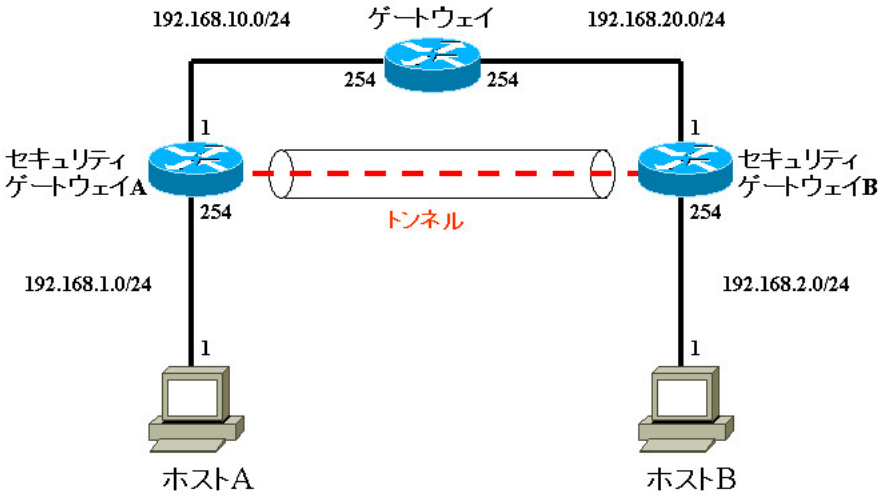

図 5.1: マシン構成

ゲートウェイは、 Cisco ルータを使用する。残りのマシンは、 FreeBSD4.10 を使用する。また、ゲートウェイでのフィルタリング条件は、表 5.1 の通りである。フィルタリング条件は、上から順番に照合していく。また、図 5.2 は、今回使用する Cisco ルータでのアクセスリスト (リスト名： ipsec) での表現である。

表 5.1: フィルタリング条件

プロトコル	送信元アドレス / ポート	宛先アドレス / ポート	可・不可
TCP	すべて / すべて	すべて /23	不可
UDP	すべて / すべて	すべて /1024 以上	不可
IP	すべて / すべて	すべて / すべて	可

```
CISCO#show access-list
Extended IP access-list ipsec
    deny tcp any any eq telnet
    deny udp any any gt 1023
    permit ip any any
CISCO#
```

図 5.2: Cisco における表現

5.2 プログラムの概要

プログラムを作成した環境は、表 5.2 の通りである。

表 5.2: プログラム作成環境

マシン名	セキュリティゲートウェイ A	セキュリティゲートウェイ B
OS	FreeBSD4.10	FreeBSD4.10
コンパイラ	gcc2.95.4	gcc2.95.4

この実験で使用するプログラムは、図 5.3 、図 5.4 のような処理を行う。前者は出力処理で、後者は入力処理の流れである。ただし、既に SA は確立されているという前提である。

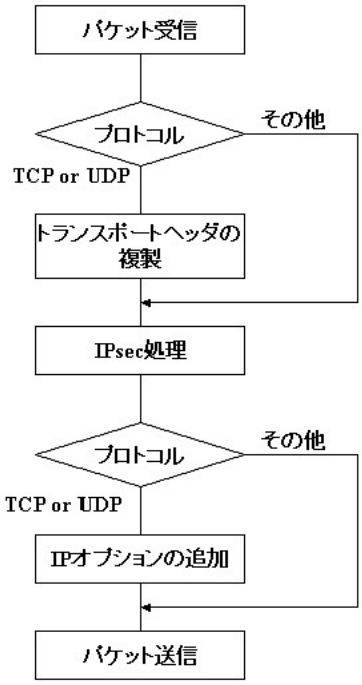

図 5.3: 出力処理

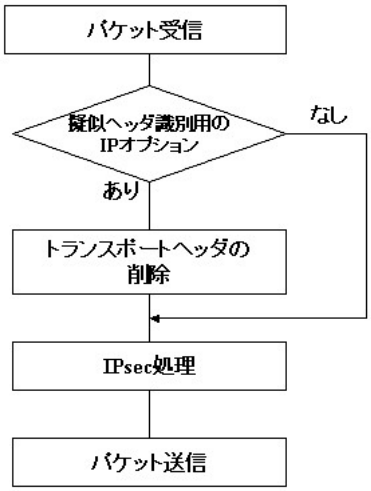

図 5.4: 入力処理

各処理のうち、提案方法に該当する部分は、出力処理の中では、「トランスポートヘッダの複製」、「IP オプションの追加」、入力処理の中では、「トランスポートヘッダの削除」である。上記の処理のうち、出力処理 (トランスポートヘッダの複製、IP オプションの追加) は約 170 行、入力処理 (トランスポートヘッダの削除) は約 130 行のコードで実装している。また、実装の際に利用した暗号、認証、圧縮の各アルゴリズムは、表 5.3 の通りである。

表 5.3: 利用したアルゴリズム

暗号	認証	圧縮
CBC-DES	HMAC-MD5-96	DEFLATE

5.3 実験結果

各フィルタリング条件を満たすようにホスト A からホスト B に対して以下の通信を行った。

- SSH (22/TCP)
- TELNET (23/TCP)
- NSLOOKUP (53/UDP)
- TRACEROUTE (1024 以上 /UDP)
- PING

5.3.1 SSH

ホスト B (192.168.2.1) に対して、ユーザ名を ipsec として、接続を試みた。図 5.5 のようにパスワードを聞かれる状態になった。ゲートウェイでは、許可しているサービスなので、相手までパケットが届いていて正常な処理がされているということである。

```
> ssh ipsec@192.168.2.1
ipsec@192.168.2.1' password:
```

図 5.5: SSH の実行結果

5.3.2 TELNET

ホスト B (192.168.2.1) に対して、接続を試みた。結果は図 5.6 のようになる。これは、ゲートウェイで不許可にしているサービスであるため、ゲートウェイでパケットが破棄されてしまうからである。また図 5.7 は、その時のゲートウェイでのログである。 telnet (23) がフィルタリングされたことがわかる。

```
> telnet 192.168.2.1
Trying 192.168.2.1...
telnet: connect to address 192.168.2.1: Connection refused
telnet: Unable to connect to remote host
```

図 5.6: TELNET の実行結果

```
#CISCO
XX:XX:XX: %SEC-6-IPACCESSLOGP: list ipsec denied tcp 192.168.10.1(50890) ->
192.168.20.1(23), 1 packet
#CISCO
```

図 5.7: ゲートウェイでの出力 (TELNET)

5.3.3 NSLOOKUP

ホスト B (192.168.2.1) に対して、ホスト A (192.168.1.1) の情報を問い合わせた。図 5.8 のような応答があった。ホスト B は、ネームサーバでないため、ホスト A についての情報をもっていない。つまり、これは、アプリケーション上のエラーであって、パケットが届いていないためのエラーではない。

```
> nslookup 192.168.1.1 192.168.2.1
*** Can't find server name for address 192.168.2.1: No response from server
*** Default servers are not available
```

図 5.8: NSLOOKUP の実行結果

5.3.4 TRACEROUTE

ホスト B (192.168.2.1) に対して TRACEROUTE を行った結果は、図 5.9 のようになった。 1 ホップ目のマシンのみ反応がある。これは、ゲートウェイで UDP パケットが破棄されるため、ゲートウェイ以降のマシンにまでパケットが届かないからである。図 5.10 は、その時のゲートウェイでのログである。 udp がフィルタリングされたことがわかる。

```
> traceroute 192.168.2.1
traceroute to 192.168.2.1 (192.168.2.1), 64 hops max, 40 byte packets
 1  192.168.1.254 (192.168.1.254)  0.218 ms  0.196 ms  0.112 ms
 2  * * *
 3  * * *
(以下略)
```

図 5.9: TRACEROUTE の実行結果

```
#CISCO
XX:XX:XX: %SEC-6-IPACCESSLOGP: list ipsec denied udp 192.168.10.1(41132) ->
192.168.20.1(33435), 1 packet
XX:XX:XX: %SEC-6-IPACCESSLOGP: list ipsec denied udp 192.168.10.1(41133) ->
192.168.20.1(33436), 1 packet
XX:XX:XX: %SEC-6-IPACCESSLOGP: list ipsec denied udp 192.168.10.1(41134) ->
192.168.20.1(33436), 1 packet
#CISCO
```

図 5.10: ゲートウェイでの出力 (TRACEROUTE)

5.3.5 PING

ホスト B (192.168.2.1) に対して TRACEROUTE を行った結果は、図 5.11 のようになった。ゲートウェイでは、許可されているサービスであるため、破棄されることなく通信できている。

```
> ping -c 3 192.168.2.1
PING 192.168.2.1 (192.168.2.1): 56 data bytes
64 bytes from 192.168.2.1: icmp_seq=0 ttl=62 time=1.312 ms
64 bytes from 192.168.2.1: icmp_seq=1 ttl=62 time=0.945 ms
64 bytes from 192.168.2.1: icmp_seq=1 ttl=62 time=0.980 ms

--- 192.168.2.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.945/1.079/1.312/0.165 ms
>
```

図 5.11: PING の実行結果

5.4 遅延時間の測定

図 5.1 の構成で、ホスト A からホスト B へ 200MB、 400MB、 800MB のサイズのファイルを送り、疑似ヘッダを使用した場合 (提案方法) と使用してない場合 (従来方法) での送信時間を測定し、遅延時間 (送信時間の差) を確かめてみた。表 5.4 は、各ファイルサイズにおける送信時間である。 200MB では 17 秒、 400MB では 39 秒、 800MB では 76 秒の遅延が生じた。尚、ファイル送信には、 SCP を使用した。

表 5.4: 提案方法と従来方法の送信時間

サイズ (MB)	提案方法 (秒)	従来方法 (秒)
200	94	111
400	185	224
800	367	443

図 5.12 は、表 5.4 をグラフにしたものである。送信時間が、ファイルサイズに比例して増加

していることがわかる。つまり、遅延時間も比例していることになる。

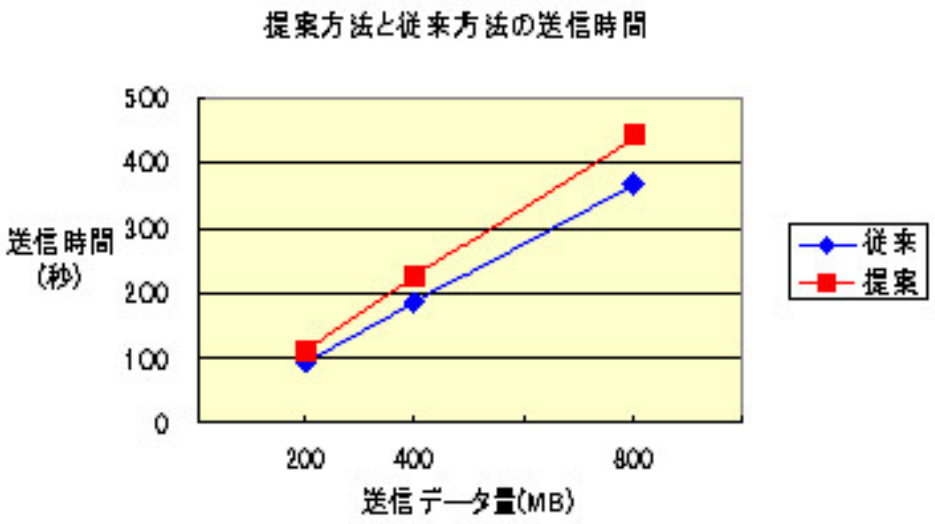

図 5.12: 提案方法と従来方法の送信時間

第 6 章

考察

アプリケーションごとの制御

前章の実行結果より、設計した通りにアプリケーションごとに制御できていることがわかる。これは今回の目的を果たしている。

遅延時間

トランスポート層のヘッダを追加するため、データの送信量が増えてしまう。前章の実験結果より、 200MB、 400MB、 800MB のサイズのファイルを送信するのに、それぞれ、 17 秒、 39 秒、 76 秒の遅延時間が発生している。しかし、実際にそれほどの大きなデータを使用する場合は少ないと考えられる。前章の実験結果より、 1MB の遅延時間を計算すると、 0.085 秒となる。仮に、よく使われる電子メールで例えると、 1MB とは、メール 1 通あたり 10KB したと仮定した場合、 100 通分に相当する (1M = 1000K)。また、 web ページの閲覧の場合を考えてみると、たいていのページは、数十 KB から数百 KB ぐらいである。仮に、 1 ページ 100KB とした場合、遅延時間は 1 ページあたり 0.0085 秒あり、気にならない時間である。このように、たいていの場合は、問題にならないと考えられる。

実装の難易度

前章で述べた通り、今回作成したプログラムにおいて、提案方式を実現している部分は、出力処理では約 170 行、入力処理では約 130 行で実装している。 IPsec は OS 実装であるため、本来なら OS (UNIX ではカーネル) にコードを書く必要がある。 OS の該当する関数に似たようなコードを書くことにより、比較的容易に実装できると考えられる。

今後の課題

今回の仕様は、 IP オプションはすべて 0 の 4 バイトのフィールドを使い、疑似 TCP ヘッダは必要なフィールド以外はすべて 0 とした。このため、ルータによっては、無効なデータとしてパケットが破棄されてしまう可能性がある。現実の環境において、どれくらいのルータを通過することができるか確かめてみる必要がある。

また、今回は SA は既に確立しているものとして実装した。今後は、従来の IPsec の方法と今回提案した疑似ヘッダを使用する方法とを SA の段階から選択できるようにすると利用しやすくなるとさらに考えられる。

謝辞

本論文を作成するにあたり、日頃より多大な御指導を頂いております後藤滋樹教授に深く感謝致します。また、日頃より有益なご意見、御指導をいただいている後藤研究室の諸氏に感謝します。

参考文献

[1] 馬場達也 著: 『マスタリング IPsec』, オライリー・ジャパン, 2001.

[2] S.Kent, R.Atkinson, RFC2401, Security Architecture for the Internet Protocol, 1998.

[3] S.Kent, R.Atkinson, RFC2402, IP Authentication Header, 1998.

[4] C.Madson, R.Glenn, RFC2403, The Use of HMAC-MD5-96 within ESP and AH, 1998.

[5] C.Madson, N.Doraswamy, RFC2405, The ESP DES-CBC Cipher Algorithm With Explicit IV, 1998.

[6] S.Kent, R.Atkinson, RFC2406, IP Encapsulating Security Payload, 1998.

[7] D.Piper, RFC2407, The Internet IP Security Domain of Interpretation for ISAKMP, 1998.

[8] D.Harkins, D.Carrel, RFC2409, The Internet Key Exchange (IKE), 1998.

[9] A.Shacham, R.Monsour, R.Pereira, M.Thomas, RFC2393, IP Payload Compression Protocol (IPComp), 1998.

[10] R.Pereira, RFC2394, IP Payload Compression Using DEFLATE, 1998.

[11] W.Richard Stevens 著, 橘 康雄 訳, 井上 尚司 監訳: 『詳解 TCP/IP Vol 1』, ピアソン・エジュケーション, 2000.

[12] Gary R.Wright, W.Richard Stevens 著, 徳田英幸, 戸辺 義人 監訳: 『詳解 TCP/IP Vol 2』, ピアソン・エジュケーション, 2002.

[13] 村山 公保 著: 『基礎からわかる TCP/IP ネットワーク実験プログラミング』, オーム社, 2001.

[14] Eric Rescorla 著: 齋藤孝道, 鬼頭利之, 古森貞 監訳: 『マスタリング TCP/IP SSL/TLS 編』, オーム社, 2003.