IPv4/v6におけるIP securityの実装

IP security implementation on 4.4BSD kernel: experiences and comments

伊藤 純一郎[†‡]　　祢宜 知孝[†]
Jun-ichiro Itoh　　Tomonori Negi

†慶應義塾大学　　‡ソニーコンピュータサイエンス研究所
Keio University　Sony Computer Science Laboratory Inc.
email: `itojun@itojun.org` / `negi@mt.cs.keio.ac.jp`

概 要

近年、より安全なインターネットの構築が急務となっている．IP層における盗聴防止とパケット毎の認証のためにipsec(IP security protocol)が提案されているが、米国の輸出規制などのために実装が広く行き渡っておらず、広域での運用には至っていない。我が国は現在のところ輸出規制等が緩いため、日本の研究者は世界に貢献できる立場にある。本稿では、BSD系UNIXカーネル内でのIPsecの実装と、実装にまつわる問題点について述べる。

1 はじめに

クラッカーによる攻撃の脅威が叫ばれて久しい。クラッカーによる攻撃方法も年々進化してきており、現在規定されているプロトコル上避け難いような攻撃、たとえばTCPにおけるTCP RST攻撃も報告されている。

これらに対抗するため、IPv4/v6のためのipパケット暗号化/認証プロトコル、ipsec(IP security protocol)がIETF ipsec working groupにより提案されている。IPv6をサポートしている全ての機器においてipsecの実装が義務づけられている[Atkinson 95c]ことからも、ipsecの必要性や重要性が伺える。にもかかわらず、ipsecの広域運用試験は、特に日本においてはほとんど進んでいない。これにはいくつかの理由があると考えられる:

1. ipsecについて主に活動しているグループの多くは米国にある。しかし、米国の輸出規制により、彼らの実装は日本国内では利用できない。
2. 日本国内で実装を行っているグループの多く(企業が多い)はエンドホスト実装を公開しておらず、ルータ製品によるVirtual Private Network運用のみが可能である。
3. 動的鍵交換プロトコルの実装は、ipsecの実装以上に少ない。

このため、WIDEプロジェクトIPv6/whips[†1] working groupの活動の一部として、国内産のipsecサポート入り4.4BSDカーネルおよび鍵交換デーモンの実装を行うこととした。我々は、多数のひとにクラッカーの脅威におびやかされることのない通信手段を提供することを望んでいる。米国外で米国産の実装が自由に利用できるように法改正がされるまでにはまだ時間がかかりそうである。対して日本では、現在暗号技術を含むフリーソフトウェアの利用および輸出入については規制がほとんどない[†2]。よって、日本から各国へ実装を提供し貢献することは重要であると考える。

本稿では我々の行った4.4BSDカーネルにおけるipsecの実装について述べる。以下ではipsecの仕様について概説したあと、実装の概略および実行性能の簡単な評価について述べる。

2 ipsecの仕様

2.1 仕様の概要

ipsecの仕様は大きくわけてAH(Authentication Header)とESP(Encapsulated Security Payload)のふたつに分けられる。rfc1826 [Atkinson 95a]で規定されるAHは、パケットが改竄/偽造されていないことを保証するためのプロトコルである。送信側でMD5などのハッシュ関数を用いて計算した暗号チェックサムをパケットに付加し、受信側で検証する。rfc1827 [Atkinson 95b]で規定されるESPは、パケットの内容が途中経路上で盗聴されることを防ぐためのプロトコルである。送信側でDESなどの暗号アルゴリズムを用いてパケット全体を暗号化し、受信側で復号することでパケットの内容を盗聴から守る。ESPにはIPパケットのペイロード部のみを暗号化するトランスポートモードと、IPヘッダをIPパケット全体を暗号化して外側に新たなIPヘッダを付加するトンネルモードのふたつがある。

AH/ESPどちらのプロトコルも、送信側と受信

[†1] WIDE Happy IP Security

[†2] Crypto Law Survey [Koops 97]によれば、暗号製品の輸出で50000円以上の対価を得る場合にのみ許可が必要。

側で暗号鍵を秘密裏に所持していることを仮定している。不特定多数の相手と ipsec を用いて通信を行う場合、鍵交換プロトコルが必要となるが、これは ipsec 自体の仕様とは独立に定義されることとなっている。[†3]

AH/ESP における各ハッシュ関数や暗号アルゴリズムの用い方については、個別の rfc が発行されることとなっている。現在のところ rfc になっているのは、AH については keyed MD5 [Metzger and Simpson 95]、keyed SHA と HMAC MD5、ESP については DES CBC モード [Karn *et al.* 95] である。

2.2 仕様の変遷

ipsec の仕様には大きくわけて 2 つある。standard track rfc である rfc1825 から rfc1829 で定義されているもの (以下旧仕様) と、ipsec working group による internet draft で定義され現在審議中の `draft-IETF-ipsec-*`(以下新仕様) がある。旧仕様と新仕様の違いは主としてリプレイアタックに対する強度の向上である。

新仕様に関しては、現在も大きな基本ドキュメントの改定が行われている。1997 年 3 月に一度ドキュメント群が準備されたが、1997 年 8 月の IETF を前にドキュメントは完全に入れ換えらた。現在の新仕様は `draft-IETF-ipsec-doc-roadmap-01.txt` にはじまるドキュメント群で定義されている。現在の新仕様は 1997 年 8 月の IETF において議論される。

実装に際し、われわれはまず旧仕様を実装し、新仕様が落ち着き次第移行する、というストラテジをとった。現在日本で入手可能な他の実装は全て旧仕様に基づいていること、新仕様が鍵交換プロトコルの存在を仮定してつくられているため先に旧仕様のカーネルの上で鍵交換デーモンを作成してから新仕様に移行した方が好ましいこと、などが理由である。

このため、本稿では特に指定しないかぎり旧仕様について取り扱っている。1997 年 8 月の IETF の結果をみて、新仕様にまつわる考察および実装の移行を行いたい。

3 実装

我々は以下の 3 プラットホーム向けの IPsec パッケージを実装、配布している:

- FreeBSD 2.2.1-RELEASE (IPv4)
- FreeBSD 2.2.1-RELEASE + WIDE Hydrangea IPv6 パッケージ (IPv4/v6)
- BSD/OS 3.0 + WIDE Hydrangea IPv6 パッケージ (IPv4/v6)

本稿の実装は、4.4BSD UNIX カーネル内の TCP/IP 処理部に対する変更として実現されている。実装の概要を図 1に示す。なお、以下では簡単のため、特に指定しない限り IPv4 上での実装について述べる。

IPv6 header nhdr = routing	routing header nhdr = AH	AH nhdr = TCP	TCP	TCP data

図 2　数珠つなぎヘッダ (例)

3.1 入力パケットの処理

パケットがインタフェースから `ip_input` に到着すると、`ip_input` ではプロトコル番号による処理のふりわけが行われる。AH および ESP にはそれぞれ固有のプロトコル番号が与えられているのでそれに従い AH および ESP の処理部へと処理が渡る。

AH の処理部では、受け取ったパケットの暗号チェックサムを計算し、パケット内に記録されている送信側で計算された暗号チェックサムの値と比較を行う。チェックサムの値が正しければ、AH のためのヘッダを剥しとり、次のプロトコル処理へ制御を移す。

ESP の処理部では、受け取ったパケットの復号化を行う。復号化の後 ESP のためのヘッダを剥しとり、次のプロトコル処理へ制御を移す。

認証や復号化に失敗した場合や、必要な暗号鍵が存在しない場合には、パケットは破棄される。認証や復号化に成功しても、届いたパケットの状態が各 TCP 接続や UDP のポート毎に定めた暗号通信ポリシに合致しなければ、パケットは破棄される。

3.2 出力パケットの処理

出力パケットに関する ipsec の処理は、`ip_output` の末尾で行われる。`ip_output` 末尾で出力すべきパケットが確定したのち、ESP、AH の順で処理を行う。ESP による暗号化が必要な場合、指定された暗号アルゴリズムと鍵管理部に登録されている鍵とを用いてパケットの暗号化を行う。AH による認証データの付加が必要な場合、指定された認証方式にもとづいてパケットに変更を加える。ESP/AH のための操作の際、暗号化や認証が必須であるのに鍵が存在しない場合、パケットは出力されずに破棄される。

3.3 上位層の修正

現在入手可能な 4.4BSD ベースの UNIX の多くでは、TCP 処理部で経路 MTU 探索の結果を考慮している。このために TCP 処理部では、TCP のデータ部に用いることのできる領域を経路 MTU と下位層のヘッダサイズから計算せねばならない。ipsec の実装を行う場合、AH や ESP のヘッダの大きさを考慮しておかねばならない。このため、TCP 処理部で「このパケットは暗号化されるかどうか」を前もって判別し、AH や ESP のヘッダ長を得て TCP データ部の長さを計算せねばならない。

3.4 IPv4 と IPv6 による違い

AH および ESP は、IPv6 で仮定されている数珠つなぎのヘッダ構造 [Deering and Hinden 95] を仮

[†3] 鍵交換プロトコルに関しては本稿では取り扱わない。

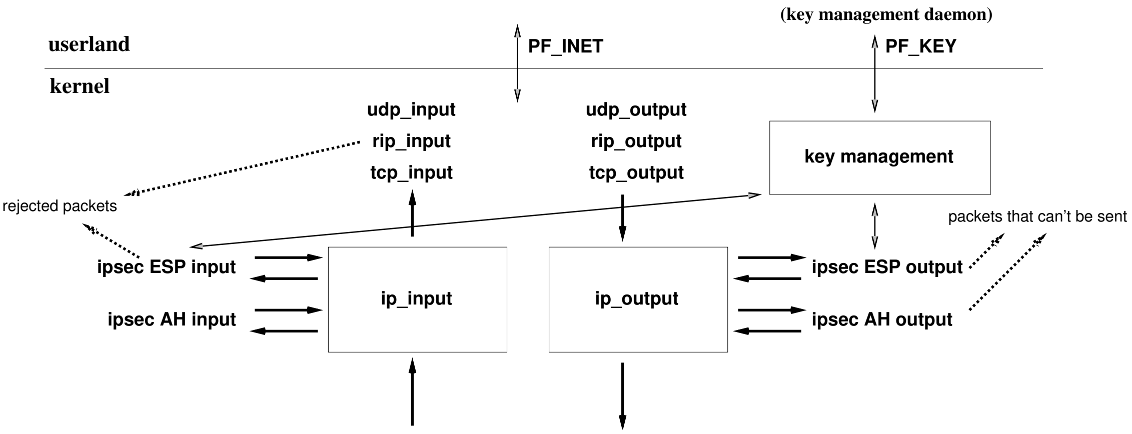

図1　実装の概要

定している (図2)。プロトコルヘッダが数珠つなぎの構造になっている場合、希望のプロトコルヘッダをみつけるためにヘッダの検索が必要になる。例えば、図2でTCPヘッダの有無を調べるためには、IPv6ヘッダから順にヘッダを追わねばならない。

4.4BSDのIPv4処理部では、数珠つなぎのプロトコルヘッダの処理に関して特に考慮をしていないため、AH/ESPの導入にともない数珠つなぎヘッダを検索する処理を追加せねばならない。IPv6の場合の処理はIPv6処理部の構成によるが、今回利用したWIDE Hydrangeaの場合特別の考慮はない。今回の実装の場合には、特に処理を汎用化せずAH/ESPのためだけに数珠つなぎヘッダの検索機構を含めてある。数珠つなぎヘッダの取扱いを高速化する方法としては、例えば以下が考えられる:

- `mbuf`の内部へのsmart pointerとして働く構造を準備し、数珠つなぎヘッダの検索回数を減らす。
- パケット受信の直後に`mbuf`を各数珠つなぎヘッダ毎に分割する。

数珠つなぎヘッダの処理の効率化は、IPv6処理部の実装の際にも重要である。今後、`mbuf`自体の改良も含めた議論を行いたい。

3.5　性能評価

本稿のFreeBSD 2.2.1-RELEASE上の実装を用い、IPv4上で旧仕様のipsecを用いftpによる性能評価を行った。ESPの暗号アルゴリズムとしてはDES CBCモード (鍵は8バイト)、AHの暗号チェックサムアルゴリズムとしてはkeyed MD5(鍵は8バイト) を用いた。測定に用いた計算機のCPUは計算機Aが486DX2-66MHz、BがAMD 5x86-100MHzである。ethernetカードはISA busもしくはPCカードインタフェース経由で接続されている。測定結果を表1に示す。

表より、ESPによるパケットの暗号化を行った場合、オーバヘッドが比較的高くなるということがわかる。試みに、ssh[†4]による通信時間も評価してみたところ、AH+ESPを用いた場合とほぼ同様の結果が得られた。これより、我々のipsec実装が(設計上の誤りなどで) 不要に遅いわけではないことがわかる。ESPを利用した場合の通信速度を高速化することが望まれる。主としてパケットの暗号化に時間がかかっているものと思われるため、今後DES処理部の見直しまたは入れ換えを検討したい。

3.6　相互接続性

1997年7月7日から4日間、WIDE IPv6/whips working groupの合宿において各参加組織によるipsec実装の相互接続性テストを行った。IPv4/v6上における旧仕様のAHやESPの相互接続性を、ping、ttcp[†5]およびtelnet等を用いて検証した。最終的には参加した全ての実装が、おのおのの通信可能な相手全てについて通信に成功した。

相互接続性試験において問題となった点を挙げておく。

- AHにおけるIPv4ヘッダの取り扱い
  rfc1826 [Atkinson 95a]では、AHの暗号チェックサム計算時にIPv4ヘッダのどこを無視して計算すべきか明確に規定していない。このため、計算される暗号チェックサムが実装により異なった。議論した結果、以下のフィールドを無視することに落ち着いた:
  > IP checksum、TTL、type of service、fragment offset、flags

  IPv6の場合、IPv6の仕様においてルータがIPv6ヘッダを書き換える可能性のある場合をすべて列挙しているため、IPv4の場合のような問題は発生しなかった。
- AH/ESPの鍵長の取り扱い
  一部の実装は鍵長が特定の長さであることを仮定していた。このため、接続性試験においては通信相手に応じてお互いに利用可能な鍵長を選択した。

相互接続性試験については別途、より詳しく発表することとしたい。

[†4] Tatu Yloenenによる暗号通信ツール。実装は`ftp://ftp.cs.hut.fi/pub/ssh/`から入手可能。

[†5] tcpおよびudpのデータ転送速度を測るツール。

表 1　ftp によるファイル転送速度 (KBytes/sec)

	ipsec モード				(参考) ssh
データ転送方向	なし	AH	ESP	AH+ESP	DES
A → B	708.2	507.6	207.8	190.6	181.2
A ← B	814.9	567.7	224.5	196.6	197.8

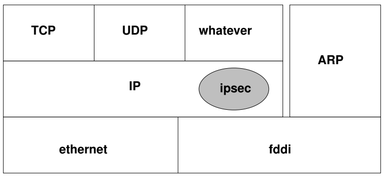

図 3　ipsec のレイヤリング

4　考察

4.1　パケット出力時のパケット破棄

本実装では、3.2節に述べたように、ipsec の処理が必須であるにもかかわらず ESP/AH のための暗号鍵が存在しなかった場合、パケットをネットワークへ出力せずに破棄している。別の実装方法としては、パケットを破棄せずにキューにつないで保存しておき、暗号鍵が登録されたときに追って出力することも考えられる。後者の場合には、キューの待ちが TCP の状態機械に与える影響や、メモリ消費に関する注意深い考察が必要になる。特に、暗号鍵が長期間にわたり登録されなかった場合、カーネル空間のメモリが枯渇することが考えられる。このため、今回の実装では前者、すなわち単純にパケットを破棄する方針を選択した。

4.2　ipsec のレイヤ

出力パケットに対する暗号化/認証の処理は、4.4BSD のネットワークコードの構造上、若干厄介である。ipsec はレイヤリング的に IP 層とほぼ同位置、または IP 層の途中に位置すると考えてよい。あるいは、ipsec は IP に対する拡張である、と理解してもよい。対して、ipsec での鍵管理は接続相手ごともしくは TCP/UDP の接続ごとに行うこととなっているため、管理情報は実装上 `socket` 構造体と関連づけることが望ましい。IP 層では本来 `socket` 構造体で管理される接続に関する情報を必要としないので、4.4BSD の実装では `ip_output` には `socket` 構造体は渡らない。

このため本実装では、`mbuf` 構造体の中の未使用フィールドに `socket` 構造体へのポインタを記録することとした。TCP や UDP などの上位層でこのポインタを設定し、`ip_output` から AH/ESP の処理を呼び出す際に引数として `socket` 構造体へのポインタを渡している。

4.3　運用上の問題

広域での ipsec 運用を行う際には、鍵交換が最大の問題となるだろう。IETF ipsec working group においては、IKMP(Internet Key Management Protocol) を標準化することが合意されている。しかし、IKMP の仕様はいまだ定まっていない。このため、一部の ipsec 実装では、IKMP の標準化を待たずに他の鍵交換プロトコルを利用している。今後の標準化過程に注目したい。

また、鍵交換プロトコルが広く利用されるようになった場合、鍵交換のための時間による TCP セッション確立時間の延びなどが予想される。早期に広域での鍵交換プロトコル運用試験等を行い、これらの影響を考察したい。

5　まとめ

本稿では、4.4BSD UNIX カーネルでの ipsec(IP security protocol) の実装に関して述べた。我々の実装は IPv4/v6 の両方をサポートし、現在安定して稼働している。今後動的鍵交換関連の整備を行い、広域運用試験を通して ipsec の実用性を考察したい。

鍵交換に関しては別途祢宜から報告があるはずなので [祢宜ほか 97] 参照されたい。

参考文献

[Atkinson 95a]　R. Atkinson. IP Authentication Header. rfc1826, August 1995.

[Atkinson 95b]　R. Atkinson. IP Encapsulating Security Payload (ESP). rfc1827, August 1995.

[Atkinson 95c]　R. Atkinson. Security Architecture for the Internet Protocol. rfc1825, August 1995.

[Deering and Hinden 95]　S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 1883, December 1995.

[Karn *et al.* 95]　P. Karn, P. Metzger, and W. Simpson. The ESP DES-CBC Transform. rfc1829, August 1995.

[Koops 97]　Bert-Jaap Koops. Crypto Law Survey version 9.4. `http://cwis.kub.nl/~frw/people/koops/lawsurvy.htm`, July 1997.

[Metzger and Simpson 95]　P. Metzger and W. Simpson. IP Authenticaiton using Keyed MD5. rfc1828, August 1995.

[祢宜ほか 97]　祢宜知孝, 伊藤純一郎, 所真理雄. 広域 IPsec のための鍵交換デーモンの実装と評価. 日本ソフトウェア科学会 第 14 回大会, Sep. 1997.