広域 IPsec のための鍵交換デーモンの実装と評価

Implimentation and evaluation of Key Management Daemon for IPsec

祢宜 知孝[†] 伊藤 純一郎[††] 所 真理雄[†††]
Tomonori NEGI Jun-ichiro ITOH Mario TOKORO

[†]慶応義塾大学大学院理工学研究科
[††]慶応義塾大学大学院理工学研究科/Sony CSL
[†††]ソニー株式会社 IT 研究所

概 要

近年，クラッカーによるインターネット上でのアタックが多く報告されており，盗聴やなりすましに強い安全なネットワークの構築が急務となっている．現在，安全なネットワークを IP レベルで提供すべく **IPsec** [3] [1] [2] が提案されているが，未だ広域での運用には至っていない．そこで本稿では，IPsec の広域での運用に必要不可欠な鍵交換デーモン (**Key Management Daemon**) を **ISAKMP(Internet Security Association and Key Management Protocol)** [5] [4] [8] に基づいて実装し，評価を行う．そして，ISAKMP の問題点を指摘し，改善策を述べる．

1 はじめに

近年 IP 層でセキュリティを確保しようとする IPsec の研究が盛んに行われている．IPsec は，図1に示したように 3 つの構成要素からなる．その構成要素とは，IP パケットの認証 [1]，暗号 [2] を行う **IP Protocol** 部と，認証や暗号を施す際に必要な情報 (SPI(Security Parameter Index)) [3] を管理する **Key Database** 部と，通信に先だって通信先と SPI を確立する鍵交換デーモン (**Key Management Daemon**) 部の 3 つである．そして，SPI が確立されている 2 者間の関係を **SA(Security Association)** と言う．

また，IP Protocol と鍵交換デーモン間のインターフェイス (図 1中 (1)) は，A Simple IP Security API Extension to BSD Sockets [6] で，鍵交換デーモンと Key Database 間のインターフェイス (図 1中 (2)) は，PF_KEY Key Management API [7] で定められている．

そして IPsec には以下に示すような 5 つのセキュリティレベルがある．

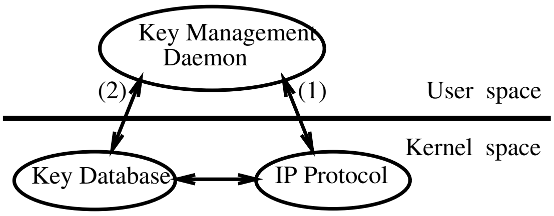

図 1 **IPsec の構造**

None

送信するパケットに対して一切のセキュリティ処理を施さず，また受信するパケットはセキュリティ処理を施されていないものでも良い．

If available

既に SA が確立されている場合には，送信するパケットに対してセキュリティ処理を施す．SA が確立されていない場合には，鍵交換デーモンに対して SA の確立を要求し，SA が確立されるまでは，セキュリティ処理を施さずにパケットを送信する．一方，受信するパケットはセキュリティ処理を施され

表 1　鍵交換プロトコル一覧表

プロトコル	対象	通信	処理
ISAKMP	IPsec SSL etc	複数回	複雑
Photuris	IPsec	複数回	単純
SKIP	IPsec	–	単純

ていないものでも良い．

Use

基本的には If available と同様であるが，SA が確立されていない場合には，鍵交換デーモンに対して SA の確立を要求し，SA が確立されるまでパケットを送信しない．

Require

送信するパケットも受信するパケットも必ずセキュリティ処理が施されていなければならない．

Require unique

基本的には Require と同様であるが，必ずセッション毎に SA を確立しなおさなければならない．

2　鍵交換デーモンについて

2.1　鍵交換プロトコル

現在，鍵交換のプロトコルとして ISAKMP [5] [4] [8] や Photuris，SKIP 等が提案されている．ISAKMP は，IPsec に限らず鍵交換を必要とするアプリケーション全てに対して鍵交換のサービスを提供できるが，鍵交換デーモン間で複数回の通信が必要であり，かつ処理が複雑であるという特徴を持っている．一方 Photuris は，鍵交換のサービス対象が IPsec のみであり，鍵交換デーモン間で複数回の通信が必要であるが，処理が比較的単純であるという特徴を持っている．SKIP は，鍵交換のサービス対象が IPsec のみであり，鍵交換デーモンを必要とせず，処理が非常に単純であるという特徴を持っている．以上の各プロトコルの特徴を表 1に示す．そして，現在 ISAKMP が世界標準の鍵交換プロトコルになりつつある．

2.2　ISAKMP

以下では，ISAKMP について簡単に説明する．ISAKMP は 2 つの mode からなる．一つは **Main Mode** であり，両鍵交換デーモン間で使用する鍵を確立し，互いに認証をおこなう．もう一つは **Quick Mode** であり，Main Mode で確立した鍵を利用して，実際に IPsec 等が使用する鍵を確立する．

さらに，Main Mode は 3 段階から構成されている．第 1 段階では Main Mode 中で使用する暗号化アルゴリズムやハッシュ関数を取り決める．第 2 段階では，鍵を確立するために必要な秘密情報や変数を交換する．そして，秘密情報や変数を用いて第 1 段階で取り決めたアルゴリズムや関数を元に鍵を生成する．第 2 段階で交換する変数は，第 1 段階で取り決めたアルゴリズムや関数に特化したものである．ISAKMP では秘密情報を交換するための方法が 3 つある．1 つ目は **Diffie–Hellman** 鍵交換アルゴリズムを利用する方法である．2 つ目は，公開鍵暗号アルゴリズム (RSA 等) を利用する方法である．3 つ目は予め offline で共有しておいた秘密情報を用いる方法である．第 3 段階では，両鍵交換デーモン間で相互認証を行う．

ISAKMP では，Main Mode，Quick Mode を通して，暗号化アルゴリズムやハッシュ関数を利用するが，利用できる暗号化アルゴリズムやハッシュ関数を限定していない．これは，ISAKMP では DES や IDEA，RC5 といった暗号化アルゴリズムや MD5 や SHA といったハッシュ関数を自由に選択することが可能であることを意味する．

Main Mode では，6 回 (3 往復) の通信が行われ，Quick Mode では 3 回 (1.5 往復) の通信が行われる．よって，ISAKMP では 1 つの SA を確立するのに，9 回 (4.5 往復) の通信が行われる．

また，ISAKMP では Main Mode の代わりに **Base Mode** や **Aggressive Mode** も提供されている．そして，Main Mode で 6 回の通信を必要としているところを，Base Mode では 4 回，Aggressive Mode では 3 回の通信で行うことが可能である．しかし，Base Mode や Aggressive Mode では通信回数が少なくてすむかわりに，ホストのなりすましの危険性をはらんでいる．

3　実装

本稿では，ISAKMP 鍵交換プロトコルに基づいた鍵交換デーモンを実装する．IP Protocol と鍵交換デーモン間のインターフェイスは，A Simple IP Security API Extension to BSD Sockets に，鍵交換デーモンと Key Database 間のインターフェイスは，PF_KEY Key Management API, Version

表 **2**　測定結果

	Main	Quick	Total
If Available	30.0 sec	6.1 sec	36.1 sec

1 に従い実装を行う．現在すでに PF_KEY Key Management API, Version 2 が提唱されているが，現行の FreeBSD のカーネルが Version 1 に準拠しているため，PF_KEY Version 1 に従って実装を行う．

4　評価

現在，鍵交換デーモンを実装中であり，現実装段階では両鍵交換デーモン間で秘密情報を交換する方法として，予め offline で共有しておいた秘密情報を用いるものに限られている．また，利用できる暗号化アルゴリズムやハッシュ関数についても限定している．

いずれ全ての機能をサポートした鍵交換デーモンを実装するが，本稿では，現実装段階での評価を行う．

以下に評価環境を述べる．

- Pentium–200MHz，Memory 64MByte，FreeBSD – 2.2.2R 2 台
- 10BaseT Ethernet の IPsec 実験専用 LAN
- IP Protocol: IPv4
- 鍵交換デーモン間で鍵を確立する方法として，予め offline で共有しておいた鍵を用いる方法を採用
- 利用できる暗号化アルゴリズム: DES–CBC
- 利用できるハッシュ関数: SHA
- IPsec レベルは If available

以上の環境の下，ISAKMP の Main Mode にかかった時間，および Quick Mode にかかった時間を測定した．結果を表 2に示す．

5　考察

ISAKMP は，さまざまなアプリケーションに対してサービスを提供することが可能である，柔軟性に富んだ鍵交換デーモンではあるが，柔軟性を確保するがために複数回の通信と複雑な処理が要求される．そのため，SA を確立するのに時間を要する結果となっている．

表 2に示したように，Main Mode で 30 秒，Quick Mode で 6.1 秒かかっている．IPsec を Use や Require，Require unique にした時には，以前に通信したことのないホストと通信する際には，最低 36 秒程は何も応答がない状態になる．しかも，Require unique レベルにした時には，通信毎に必ず最低 6 秒程無応答状態が起こる．Main Mode で 30 秒，Quick Mode で 6 秒の処理の大部分は，ISAKMP が複雑な処理を必要とすることに因る．

表 2に示した値は，ISAKMP がネットワークと CPU を占有できる環境下での値であり，実際に ISAKMP を運用する際にはより多くの時間が必要となる．これは，実際に鍵交換デーモンを運用していく上で，大きな障害となる．

6　問題点

以上より ISAKMP の問題点として，主に以下の 3 つの点が挙げられる．

- 通信負荷
- サーバへの負荷
- 認証局等との関係

以下に，それぞれの問題点の詳細を述べる．

6.1　通信負荷

ISAKMP では，Main Mode で 6 回 Quick Mode で 3 回と非常に通信回数が多い．telnet や ftp のような一度通信したホストと何回も通信をする場合，最初に SA を確立するのに時間がかかるだけで，その後の SA の確立は比較的容易である．しかし，http のような数回の通信を多数のホストと行う場合には，ホストが異なる毎に Main Mode と Quick Mode を経るので，時間がかかると同時に総通信量が増え，またネットワークへの負荷が無視できなくなる．

6.2　サーバへの負荷

ISAKMP の処理が複雑であるため，多数のホストからアクセスされるサーバには非常に負荷がかかる．そして，多数のホストに対する SA を記憶しておかなければならず，巨大な容量のメモリやハードディスクを必要とする．

6.3　認証局等との関係

ISAKMP では，Diffie–Hellman の公開値や公開鍵暗号における公開鍵が正当なものであることの認証を **CA(Certification Authority)** や **SDNS(Secure Domain Name System)** 等に依存している．例えば，CA を利用する際には，

CA にアクセスをし，認証書を発行してもらわなければならず，また Main Mode 中に通信相手から送信されて来る認証書をもとに認証を行わなければならない．従って，効率良く認証処理を行わないと，SA を確立するための時間がさらに増えてしまう．

7　解決方法

本節では第 6節で述べた問題点に対する解決方法について述べる．

7.1　通信負荷に対する解決方法

現在の ISAKMP では Main Mode における通信が 6 回である．しかし，プロトコルを変更することで，Base Mode や Aggressive Mode のようにホストのなりすましの危険を伴うことなく，4 回に減らすことが可能である．ISAKMP では，第 1 段階と第 2 段階で送信側から受信側へ 2 回，受信側から送信側へ 2 回の計 4 回の通信を経て，鍵を確立するのに必要な情報をやり取りしている．しかし，この 4 回の通信をそれぞれ 1 回づつ，計 2 回の通信で行うことが可能であると思われる．

現在は，始めに利用するアルゴリズムを決め，その後にそのアルゴリズムに特化した変数のみを送信している．しかし，ここで通信回数を減らすために，始めに利用可能なアルゴリズム全てに対応する変数を一度に全て，送信してしまう．つまり，送信側は最初の通信の際に，利用可能なアルゴリズムとそれらに特化した変数を全て一度に送るのである．そして，受信者は，選択したアルゴリズムに特化した変数を送信するのである．

上記のように通信回数を減らすことが可能ではあるが．この時 UDP のパケットサイズが問題となってくる．一度に大量の情報を送信するために，UDP のパケットサイズを越えてしまう可能性がでてくる．IPv6 では UDP のパケットサイズの最大が任意であるが，IPv4 では UDP のパケットサイズの最大が決まってしまっている．そこで，IPv4 では送信側が好ましいと思う順に，利用可能なアルゴリズムとそれに特化した変数の組み合わせを，UDP のパケットサイズの限界まで詰め込み送信することで，パケットサイズの問題は解決可能である．

7.2　サーバへの負荷に対する解決方法

ISAKMP では，処理の複雑さがセキュリティを確保する一つの要因となっている．従って，処理を単純にすることは望ましくない．サーバへの負荷の軽減を実現する方法として，ハードウェア化が考えられる．ハードウェア化の解決方法以外には，ハイエンドのマシンをサーバとし，巨大な容量を擁したメモリとハードディスクを設置する以外に解決策はない．

8　結論

本稿では，まだまだ不完全ではあるが，ISAKMP に基づき IPsec に必要不可欠な鍵交換デーモンを実装し，評価を行った．近い内に，より完全なものに近づけた鍵交換デーモンをリリースする予定であり，是非皆さんに使って頂きたいと考えている．

また，鍵交換デーモンの問題点を指摘し，その解決方法を述べた．しかし，本稿では明確な認証局等との関係に対する解決方法を提案できなかったので，今後検討していく．さらに，鍵交換デーモンを実際に運用した上で SA の確立にかかる時間を計測すると共に，本稿で提案した解決方法を実装し，評価・比較を行っていきたいと思う．

参考文献

[1] R. Atkinson. *IP Authentication Header.* Technical Report, *Request for Comments 1826*, August 1995.

[2] R. Atkinson. *IP Encapsulating Security Payload (ESP).* Technical Report, *Request for Comments 1827*, August 1995.

[3] R. Atkinson. *Security Architecture for the Internet Protocol.* Technical Report, *Request for Comments 1825*, August 1995.

[4] D. Carrel D. Harkins. *The resolution of ISAKMP with Oakley <draft–ietf–ipsec–isakmp–oakley–03.txt>.* Technical Report, *IPSEC Working Group INTERNET–DRAFT*, February 1997.

[5] Douglas Maughan, Mark Schertler, Mark Schneider, and Jeff Turner. *Internet Security Association and Key Managament Protocol(ISAKMP) <draft–ietf–ipsec–isakmp–07.txt>.* Technical Report, *IPSEC Working Group INTERNET–DRAFT*, February 1997.

[6] D. L. McDonald. *A Simple IP Security API Extension to BSD Sockets.* Technical Report, *Network Working Group Internet Draft*, March 1997.

[7] D. L. McDonald, C. W. Metz, and B. G. Phan. *PF_KEY Key Management API, Version 2 <draft–mcdonald–pf–key–v2–02.txt>.* Technical Report, *Network Working Group Internet Draft*, 1997.

[8] Derrell Piper. *The Internet IP Security Domain of Interpretation for ISAKMP <draft–ietf–ipsec–ipsec–doi–02.txt>.* Technical Report, *Network Working Group Internet Draft*, February 1997.