セキュアネットワーク構築術

白橋 明弘（ネットワンシステムズ（株））

1999 年 12 月 16 日

Internet Week 99　パシフィコ横浜

（社）日本ネットワークインフォメーションセンター編

目次

1 概要

この講演では、各種セキュリティ技術を組み合わせてネットワークを設計したり構築したりするときに考慮しなければならない課題や、解決しなければならない問題を、実務経験から得られたノウハウを交えて説明します。

具体的には、次のセキュリティ技術を取り上げています。

- ファイアウォール（2 を参照）
- 認証（3 を参照）
- リモートアクセス（4 を参照）
- VPN（5 を参照）

2 ファイアウォール

ファイアウォールを導入するためには、製品自体や TCP/IP だけでなく、アプリケーションに関してまでの幅広い知識が必要となります。また、インターネット利用環境にファイアウォールを導入するときには、DNS やメールに関する知識も必要です。ファイアウォール製品は、セキュリティのための設定作業を容易にし、誤りの発生を減らしてくれます。しかし、それ自体でファイアウォールの導入に伴うシステム構築の問題を自動的に解決してくれるわけではなく、セキュリティポリシーが適切に実現されることを保証してくれるわけでもありません。このため、管理者は、ネットワークの利用目的を把握し、何のために何を実施するのかを確実に理解して、ファイアウォールを構築していく必要があります。

また、ファイアウォールというものには、通信の中継を行う上で本質的な限界があります。これは、ファイアウォールがエンドノード間の通信の途中に立ちはだかり、不完全な情報しか得られない状況で、通信のコンテキストを記憶・分析するための資源も限られた状況で、通信を中継しなければならないためです。また、アプリケーション側が RFC 等の規格に正しく従った実装をしていないため、ファイアウォールでうまく処理できない場合もあります。このように、実際的なファイアウォールの導入にはさまざまな苦労が伴います。

2.1 トラブル事例

ここでは、ファイアウォールの導入に伴って発生した FTP 関連のトラブル事例を紹介します。ご存じのように、FTP では次の 2 種類の TCP 接続が使用されています。

- コントロールコネクション（port 21）
- データコネクション（port 20）

データコネクションの張り方には、PORT モードと PASSIVE モードという 2 種類があります。PORT モードでは、サーバ側のポート番号 20 からクライアント側の非特権ポート（≧ 1024）に接続が張られます。クライアントが PASSIVE モードでの接続を要求したときには、クライアント側の非特権ポートからサーバ側の非特権ポートに対する接続が張られます。

どちらの接続モードの場合も、コントロールコネクションの接続後、そこでやりとりされた内容に依存してデータコネクションの接続が確立されるため、ルータで行うような単純なパケットフィルタリングでは、FTP をうまく処理することができません。このため、一般的なファイアウォール製品では、クライアントからサーバへの接続においてアプリケーションの内容を監視し、対応するサーバ側からのデータコネクションを必要なときだけ特別に許可するようになっています。

2.1.1 トラブル事例 1

最初に紹介するトラブル事例は、ファイアウォールを経由したときに、特定の FTP サーバとの間で PORT モードのデータコネクションが確立できないというものです。ただし、このケースでは、PASSIVE モードでは問題は発生しないことがわかっていました。

このトラブルの原因は、FTP サーバがデータコネクションに使用するソースポート番号が RFC で標準とされている 20 ではなく、1024 以上の非特権ポートであったことで、ファイアウォールがこのソースポートを厳格にチェックしていたため、データコネクションの接続が通過できなかったのです。このような RFC に準拠しない実装の FTP サーバは、Windows 環境で動作するもの等に見られます。

2.1.2 トラブル事例 2

2 番目に紹介するトラブル事例は、ファイアウォールを経由して FTP サーバに接続すると、FTP サーバからの最初の Welcome メッセージすら表示されないというものです。

原因を究明するためにパケットダンプを取得してみると、ファイアウォールからポート番号 21 で FTP サーバに接続したときに Welcome メッセージが返されているのですが、このメッセージがファイアウォールからクライアントに中継されていませんでした。これは、通常の各 ftp コマンドとその応答の処理では行末コードが CR+LF であるのに対して、この FTP サーバが返す行末コードが LF のみであったため、ファイアウォールが行末コードとして認識できずに待ち続けていたのが原因でした。

2.1.3 トラブル事例 3

3 番目に紹介するトラブル事例は、ウィルス対策ゲートウェイとパケットフィルタリング方式のファイアウォールの組み合わせにおいて、外部の FTP サーバに対してログインした直後に、そのセッションが切断されてしまうというものです。

このトラブルの原因は、組み合わせたウィルス対策ゲートウェイが PORT コマンドを「PORT」という文字列と、その後に続く IP アドレスやポート番号の文字列を、分割した IP データグラムとして送っていたことでした。通常、このように分割されていても、上位層の TCP データストリームとしては連続したものとなるため、アプリケーション上は問題は発生しません。しかし、このとき使用していたファイアウォールはパケットフィルタリング方式のもので、前述したように利用できる資源が有限である以上やむを得ないことですが、TCP データストリームを完全に復元することはできません。このファイアウォールでは、最初の「PORT」のパケットを受け取ったところで、その内容を判断する実装になっていたため、これを不正な PORT コマンドとして、セッションを切断してしまっていたわけです。

2.1.4 トラブル事例からの教訓

前述のトラブル事例から、次の教訓を得ることができます。

- 世の中にはRFC等に従った常識的な実装とはなっていないアプリケーションが存在する一方、一般にファイアウォールでのプロトコルのチェックは厳格なことが多いため、問題が発生することがある。
- プロトコルで規定されているコマンドを独自に拡張しているような「もどき」アプリケーションは、ファイアウォールでの対応の可否について確認が必要である。
- パケットフィルタリングのファイアウォール等で使われる、上位層の内容を下位層でチェックするアプローチには本質的な限界がある。

2.2 アプリケーションへの対応

一般的でないアプリケーションをファイアウォールを越えて使う必要がある場合は、その対応の可否を判断しなければなりません。TCP/IP のアプリケーションだから問題ないと考えるのは全くの間違いです。したがって、ファイアウォールを導入する場合には、原則として利用するすべてのアプリケーションをリストアップし、問題がないことを事前に確認する必要があります。

このような手間は、パケットフィルタリング方式でも、プロキシを利用するアプリケーションゲートウェイ方式のファイアウォールでも基本的には変わりません。セキュリティを無視してまでも接続を可能にしたいときには、パケットフィルタリング方式のほうが対応しやすいこともあります。ただし、パケットフィルタリングに加えて NAT（Network Address Translation）によるアドレス変換機能を利用した場合、利用できなくなるアプリケーションもあるので注意が必要です。

ただし、ファイアウォールの対応リストに載っていないアプリケーションでもその内容によっては、ファイアウォールのルールの設定によって対応することができます。この可否の判断基準としては、一般的に以下のような点が、確認すべきポイントとなります。これらの条件を満たす「単純な」プロトコルならば、確実とは言えないまでも、対応できる可能性がかなりあると言えます。

- 接続先ポート番号が固定されている。
- 接続はクライアント側からサーバ側に確立される。
- サーバ側からクライアント側に張り返される接続がない。
- ソースの IP アドレスが変換されても問題が発生しない。
- ソースのポート番号が変換されても問題が発生しない。
- アプリケーションが扱うデータ内にIPアドレスやポート等の情報がない。

2.3 プロキシ方式の分類

パケットフィルタリング方式によるファイアウォールは、その設定が見た目は直観的で比較的わかりやすいものとなっています。実際にきちんと設定を行うには、プロトコルに関する知識が必要な場合もあり、必ずしも簡単ではありません。これに対して、アプリケーションゲートウェイ方式で使用されるプロキシには、機能的にいくつかの方法が存在し、その設定にアプリケーションの知識も必要になったりするため、難しいと思われがちです。しかし、ネットワーク構成に合った適切な導入を行えば、パケットフィルタリング型のファイアウォールだけでは実現できない機能を、効率良く実現できるというメリットがあります。

プロキシサーバを利用したときには、クライアントは直接にはプロキシサーバと通信することになります。このため、IP パケットの宛先アドレスは、基本的にプロキシサーバの IP アドレスとなります。ただし、プロキシサーバは受け取ったリクエストを最終的なサーバに送らなければならないため、何らかの方法で、本来クライアントが通信したい相手の IP アドレスを知る必要があります。このための方法によって、次のようにプロキシを分類することができます。

- 固定で割り当てる。
- 対話形式で入力させる。
- プロトコル中に組み込む。
- 透過的に処理する。

このうち、固定で割り当てる方法は、プロキシサーバの特定の IP アドレスとポートへの接続を、特定の宛先アドレスに中継するものです。たとえば、DeleGate の tcprelay や udprelay がこの方式の中継です。

また、対話形式で入力させる方法では、ユーザはいったんプロキシサーバにアクセスした後、プロキシサーバからの入力要求に対して実際に接続したい相手の IP アドレスやホスト名を指定します。

プロトコル中に組み込む方法は、プロトコル自体でプロキシのしくみが提供されているときのみ利用でき、接続先の IP アドレスを渡すものとホスト名を渡すものに分かれます。

このうち、IP アドレスをゲートウェイであるプロキシサーバに渡す方法は、たとえば SOCKS プロキシで使用され、図 1 に示すように、プロキシサーバを宛先とする IP パケットのデータ中に、目的のサーバの IP アドレスが埋め込まれて渡されます。

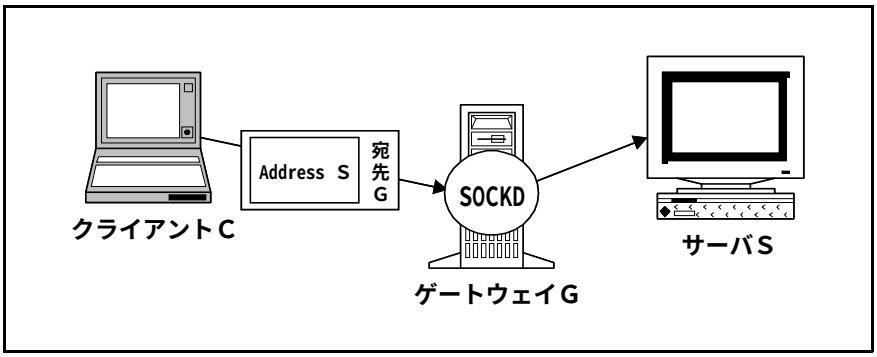

図 1：SOCKS による方法

また、ホスト名をプロキシサーバに渡す代表的な方法には HTTP プロキシがあります。HTTP プロキシでは、図 2 に示すように、プロキシサーバに対して、目的のサーバのドメイン名が渡されます。

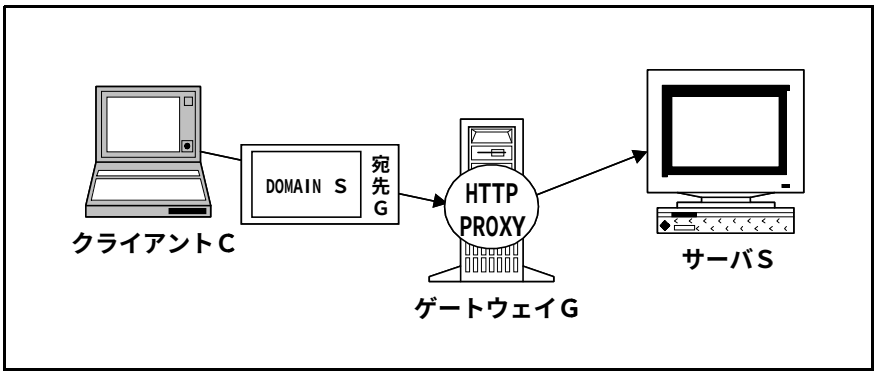

図 2：HTTP プロキシによる方法

最後に示した、透過的に処理する方法では、図 3 に示すように、クライアントは接続先のサーバの IP アドレスをそのまま使用します。

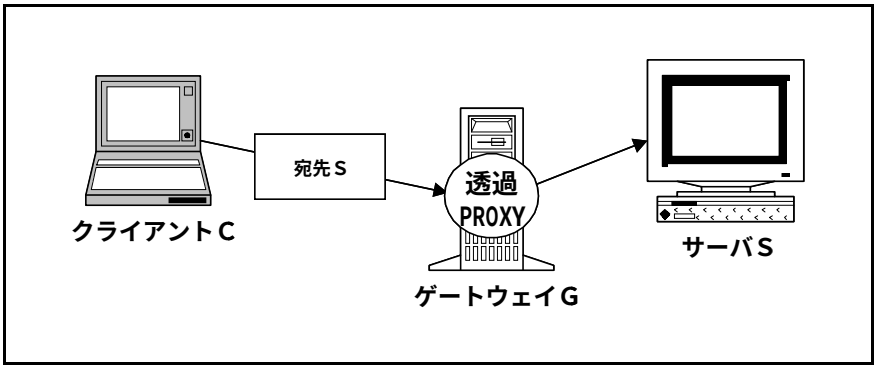

図 3：透過的プロキシによる方法

このような透過的プロキシによる方法では、デフォルトルートがファイアウォールとなるように適切にルーティングを構成しておき、ファイアウォールでパケットを検出して、プロキシプロセスを自動起動させます。また、このためには、ゲートウェイとなるサーバのオペレーティングシステムで、自分宛でない IP アドレスのパケットを処理するしくみをカーネルに組み込んでおくようにします。そして、自動起動されたプロキシプロセスによって、クライアントが望む相手先への実際の接続が実行され、クライアントとの通信が中継されます。

このような透過的プロキシの利用が、現在の商用ファイアウォール製品では主流となっています。透過的プロキシによる利点は、次のとおりです。

- クライアントの設定変更が必要ない。
- 明示的にはプロキシを利用できないプロトコルにも使用できる。

また、透過的プロキシは、組織内からインターネットに接続する部分のファイアウォールで使われるのが最も一般的ですが、イントラネットのファイアウォールの場合にも有用です。

ただし、透過的プロキシを利用するときには、次の点に注意する必要があります。

- 接続先 IP アドレスへのルーティングを適切に設定する。
- ドメイン名で接続先を指定するときには、クライアントで DNS 解決ができるようにする。
- ユーザ認証やコンテンツフィルタリングと組み合わせたときには、いくつかの制限が生じることがある。

イントラネットやエクストラネットでは、技術上や管理上の制約によって適切なルーティングを設定できず、透過的プロキシを利用できないことがあります。このようなときには、リダイレクションプロキシ／リバースプロキシ（プロキシの場合）、スタティックマップ（NAT の場合）と呼ばれる固定割り当てによる方法を利用するようにします。これらの方法では、クライアントはファイアウォールの内側の IP アドレスに接続し、その接続をファイアウォールが接続先サーバの IP アドレスに変換して中継します。

セキュリティには直接関係しませんが、透過的プロキシを応用した透過的キャッシュという方法が利用され始めています。通常、Web システムに対するプロキシとしては、前述の HTTP プロキシが利用されています。このときには、HTTP プロキシを利用するために Web ブラウザ上でプロキシの IP アドレスやドメイン名を指定しますが、これらの情報を多数のユーザに正しく設定させるためには大変な労力が必要となります。この問題を解決するために、透過的キャッシュが利用されています。

透過キャッシュを利用することで、ユーザは Web ブラウザ上で明示的にプロキシを設定する必要がなくなります。ただし、このときには、ポート番号 80 のみが対象となり、レイヤー 4 スイッチ、ポリシールーティング、WCCP、IP フィルタ等によって、トラフィックを横取りするしくみが新たに必要となります。また、透過キャッシュの利用では、次の点にも注意する必要があります。

- 一般に、ポート 80 以外の Web サーバに対するアクセスはキャッシュの対象とならない。キャッシュされないこと自体はあまり問題ないとしても、URL Filtering や Virus Check といった機能と組み合わせている場合は、これがチェックの対象外となってしまう可能性がある。それが困るなら、ポート 80 以外の Web の利用を禁止しなければいけないかもしれない。
- インターネット接続の出口に置かれたファイアウォールが http プロキシとして振る舞う場合、また、前述の URL Filtering や Virus Check のプロキシがある場合、透過型キャッシュ装置から、そうした上位の明示的なプロキシとのカスケード（多段）接続が可能かどうかが問題となる。

2.4 NAT/ アドレス変換

NAT（Network Address Translation）は、元々は RFC1631 で規定されている 1 対 1 のアドレス変換技術を指す用語でしたが、最近ではむしろパケットレベル（ネットワーク層）でのアドレス変換技術の総称として使われるようになっています。これに対して、ポート変換を伴うアドレス変換技術の呼称（用語）には決定版がなく、NAPT（Network Address and Port Translation）、IP masquerade、PAT（Port Address Translation）等が使われているため、その内容の把握に注意が必要です。

呼び方はともかくとして、NAT のアドレス変換には、基本的に次の 3 種類のモードがあり、それぞれ機能や特性が異なっています。

- 1 対 1 の静的な変換
- アドレスプールからの動的なアドレスの割り当て
- 1 つのアドレスをポート変換によって多重化した割り当て

NAT 機能を用いたファイアウォールは、一般に次のようなポリシー要件がある場合、典型的には大学等の教育研究機関での使用に適していると考えられます。

- 内部から外部へのアクセスに関しては、ポリシー上の制限は設けない。
- サーバを管理するスキルがある。
- 高速な対外接続を持っている。

NAT に対する注意点としては、NAT は「アドレス変換」という言葉の印象よりも、実際にはかなり解りづらい技術であるということがあります。たとえば、同じ Cisco Systems 社の製品である、Cisco ルータの IOS ソフトウェアの NAT 機能と、PIX Firewall の NAT 機能では、考え方や実現できる機能に大きな差があります。このため、複雑な設定を行う際には、各製品の実装仕様に関する詳細な知識が必要になる場合があります。

2.5 ファイアウォールに関係するいくつかの注意点

2.5.1 IDENT 問題

バージョン 8.8 以降の sendmail で標準的な sendmail.cf の設定を使用した場合、SMTP の接続要求に対して、サーバからクライアントのポート番号 113 に対して ident プロトコルによるユーザ認証が要求されます。これに対しては、ファイアウォールから ICMP destination unreachable （port unreachable）を返すことで、その後の処理が継続されます。ただし、フィルタリングによってブロックしたり、ICMP destination unreachable （host unreachable）を返したりしたときには、タイムアウトまで 30 秒程度待たされることや、セッションが切断されてしまうことがあります。また、同様の問題は、tcp_wrapper 等で ident を利用した場合にも発生することがあります。

2.5.2 DNS 問題

インターネット接続用にファイアウォールを導入したときには、DNS（Domain Name System）サーバは、内部向けのものと外部向けのものの 2 種類に分ける構成（Split DNS、Dual DNS）が良く利用されます。その場合、外部向け DNS には外部に対して公開する情報のみを登録し、内部向け DNS には内部のホストで必要な情報をすべて登録するようにします。そして、外部向け DNS と内部向け DNS は、スレーブフォワードで接続します。また、通常、内部向け DNS では、ファイアウォール以外で動作している DNS サーバをプライマリとし、ファイアウォールはセカンダリとします。これに対して、外部向け DNS では、ファイアウォールをプライマリとし、ISP にセカンダリを依頼するのが一般的です。

ファイアウォールと DNS についての問題の 1 つに、サブドメインの委託（delegation）に関するものがあります。DNS は、自らの管轄外の（authority を持たない）ゾーンについての問い合わせは、NS レコードを参照せずに、スレーブフォワード先の DNS にそのままフォワードしてしまいます。このため、ファイアウォール上の DNS サーバは、基本的に内部のすべてのサブドメインのセカンダリとなる必要があります。ただし、組織規模が大きいときには、サブドメインが増減するたびにファイアウォール上の DNS で登録を変更するのは、管理上困難な場合があります。この問題に対処するために、BIND 4.9 では No Forward Patch が用意されていますし、新バージョンの BIND 8.2 では、各ゾーンごとにフォワードを指定する機能が追加されています。

また、ファイアウォール内のプライベートアドレスの逆引きは、必ず内部向け DNS でローカルに解決できるようにしておく必要があります。これは、プライベートアドレスの逆引きをインターネットに問い合わせると、read-rfc1918-for-details.iana.net という応答が返されますし、その無駄な問い合わせによってサーバの応答が遅くなったりすることがあるからです。

2.5.3 Routing 問題

ファイアウォールを導入するときには、ルーティングにも注意する必要があります。たとえば、ファイアウォールで透過的プロキシを使用する場合、宛先 IP アドレスへのルーティングがファイアウォールに向くように設定されている必要があります。ファイアウォールでは、セキュリティ上の問題を発生させないために、積極的にダイナミックルーティングプロトコルに対応していることは少なく、経路はスタティックに設定することが多くなります。また、通常、同一インタフェース上でもパケットフォワーディングはしませんし、ICMP リダイレクトにも対応していないことがあります。このため、ファイアウォールを普通のルータのつもりで経路設計を考えると、落とし穴にはまることがありますから注意が必要です。

2.5.4 IP アドレス問題

ファイアウォール内でプライベートアドレスを利用するときには、RFC1918 で規定されている 10/8、172.16/12、192.168/16 のいずれかの IP アドレスを使用します。このとき、将来の合併、関連企業間との接続、VPN（Virtual Private Network）の導入等で IP アドレスが衝突する可能性を低くするために、プライベートアドレスは先頭からではなく、ランダムな位置から利用することが推奨されています。RFC1918 以外のローカルアドレスを使ってインターネットに接続する場合や、相互接続でアドレスが衝突して、アドレスの付け替え（リナンバリング）もできない場合は、制約はある方法ですが、プロキシや NAT を 2 段構成にして、衝突するアドレスをお互いに隠すことで通信を可能にできます。

3 認証

EDI（Electronic Data Interchange）やエクストラネットの利用拡大に伴って、組織外からのアクセスに関する要求が高まっています。このときに必要となるのが、強力な認証です。ただし、認証にはさまざまな実現方法があり、どのケースにどのような認証を用いるべきかは、そのしくみを良く理解していないと判断を下すことができません。ここでは、認証方式について整理していきます。

3.1 パスワードによる認証

まず、現在多くの認証で利用されているパスワード方式には、次の 2 種類があります。

- 再使用可能パスワード（固定パスワード）
- ワンタイムパスワード（使い捨てパスワード）

このうち、固定パスワードである再使用可能パスワードは、その内容が他人に知られると、不正に使用されてしまう危険性があります。これに対して、使い捨てパスワードであるワンタイムパスワードは、内容が知られても再利用できないため安全です。この 2 種類のパスワード方式については、「再使用可能パスワードが平文パスワードであり、ワンタイムパスワードが暗号パスワードである」と誤解されていることがあるので注意してください。また、「暗号化された固定パスワードは安全か」と言うと、仮に暗号化に鍵長 128 ビットの SSL を使っていたとしても、システムにアクセスするパスワードが 4 桁等では、認証については安全とは言えません。

ワンタイムパスワードの代表的なものに「チャレンジ & レスポンス型」があります。この方法では、ホストがクライアントにチャレンジを送り、クライアントがローカルにパスワードと組み合わせて演算した後、その結果をレスポンスとして送り返します。そして、サーバでも同一の演算を実行し、クライアントからの結果と照合することで、認証の可否が決定されます。ホストが送り出すチャレンジの代わりに時刻やカウンタを利用して、ホストとクライアント間を同期させる「同期型」という方式もあります。このようなワンタイムパスワードの代表的なものとしては、オープンな規格であるS/Key、OTPや、商用製品としてはSecurID、SafeWord等があります。

ワンタイムパスワードの利用は手間のかかるもので、特にチャレンジ & レスポンス型のものは、面倒さから敬遠されることが多いようです。この負担を軽減するため、S/Key や OTP の場合、dotkey や otpsock といった自動入力ツールが作られています。また、SecurID や SafeWord の場合、ソフトウェアベースのトークンを利用することで、カット & ペーストやダイアルアップネットワークとの連携等もできるようになります。ただし、ソフトウェアトークンは、認証情報が PC と一体化してしまうことによりセキュリティが低下してしまう可能性があるため、スマートカード等に秘密情報を格納するといった手法を合わせて利用する方法が注目されています。

3.2 認証のしくみ

クライアントとサーバ間での認証では、認証しようとするアプリケーションのプロトコルが、認証に対応していなければなりません。たとえば、プロトコルが想定している認証方式がユーザ名と固定パスワードを利用するものである場合、そこに、そのままチャレンジ & レスポンス型のワンタイムパスワードを利用することはできません。

また、認証の行われる流れとしては、クライアントからの接続を受け付けたアプリケーションのサーバは、認証に関してはクライアントとして機能し、認証サーバに対して認証要求を送ることになります。アプリケーションのサーバ機能を果たすのは、ログインするホスト、アクセスサーバ、VPN 装置、ファイアウォール、Web サーバ等になりますが、これらのサーバは認証クライアントとしての機能を持っていなくてはなりません。認証クライアントと認証サーバ間のやりとりに使われる認証プロトコルには、現状 RADIUS が最も広く使われていますが、トークンカード製品独自のプロトコルや NT ドメイン認証も一部使われていますし、将来的には LDAP 等も利用されることになるでしょう。

以上のような認証を行う動作と、それによって認証された情報（クライアントとユーザ名の対応関係等）を利用することは、まったく別のことなので注意してください。一般的には、認証を行ったサーバ上でしか、認証によって得られた情報は利用できません。たとえば、アクセスサーバで認証された情報は、別のファイアウォールや Web サーバでは利用できません。最初一度だけ認証を行えば、そのユーザの権限ですべての資源にアクセスができるようにする、いわゆるシングルサインオンを実現するためには、認証された情報を他のサーバからも参照できるようにする必要があります。現状では、これを実現するための一般的なしくみは存在せず、部分的に特定の製品において、クライアントの IP アドレスをキーとして連携させたり、Web システムの cookie 等の何らかのチケットをクライアントに持たせたりすることで実現されています。

3.3 デジタル証明書による認証

公開鍵暗号を使用する認証では、基本的には、認証を受けるクライアントはある情報を秘密鍵で暗号化してサーバに送り、サーバはそれを公開鍵で復号化して元の情報と一致するかどうかで認証を行います。このときに利用される秘密鍵はクライアントのみが保持し、公開鍵はサーバのみが保持していればよいことになります。この認証に使われる公開鍵を、その人・組織を特定する情報と一緒にしたものがデジタル証明書です。デジタル証明書には、CA（Certificate Authority：認証局）と呼ばれる機関による電子署名が施され、その正当性を検証できるようになっています。

図 4 に示す X.509 によるデジタル証明書は、ITU（International Telecommunication Union：国際電気通信連合）によって標準化されているデジタル証明書の書式です。このデジタル証明書には、ユーザ ID や公開鍵が収められています。さらに、デジタル証明書の内容全体がハッシュ関数で処理され、その値が CA の秘密鍵で暗号化されて、署名として添付されています。この署名を CA の公開鍵で復号して一致すれば、証明書の正当性が確認できるわけです。この X.509 によるデジタル証明書は、SSL（Secure Sockets Layer）/ TLS（Transport Layer Security）、S/MIME（Secure MIME）、SET（Secure Electronic Transaction）、IPSec/IKE 等のさまざまなアプリケーションで利用されています。

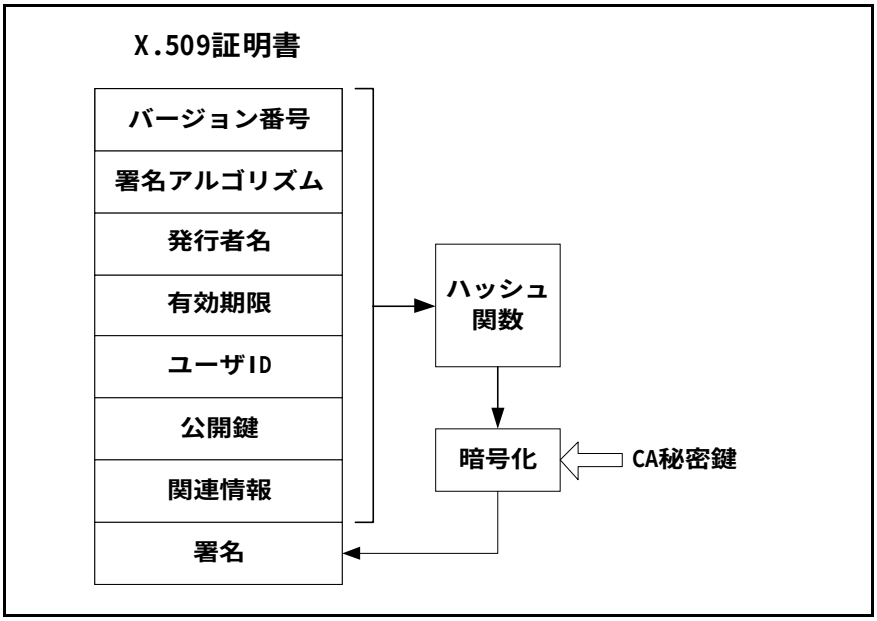

図 4：X.509 によるデジタル証明書

PKI（Public Key Infrastructure：公開鍵基盤）は、デジタル証明書をインフラストラクチャとして利用するためのしくみで、IETF（Internet Engineering Task Force：インターネット技術検討部会）の PKIX 作業グループによって標準化が進められています。現在までに、RFC2510、2511、2527、2528、2559 が規定されています。また、この標準化には、RSA Security 社の PKCS 標準や Entrust 社の提案が大きな影響を与えています。

PKI において、現在までにほぼ問題なく標準化され実装されている部分は、X.509 によるデジタル証明書の書式と、無効証明書の管理の部分です。また、証明書発行とその管理に関するプロトコルや証明書を配布するためのプロトコルについては標準化が終了していますが、個別の製品に独自の仕様が残ったままとなっているため、順次対応されていくものと思われます。さらに、既に一部の製品では、証明書の有効性チェック、CA 間での相互認証、鍵管理等も実装され始めていますが、標準化は今後の課題となっています。

なお、PKI については、その概念の分かりづらさもあって、いろいろな誤解が生じています。典型的なものを表 1 にまとめておきます。

表 1：PKI に関する誤解

誤解	正しい考え方
デジタル証明書はワンタイムパスワードよりも強力な認証である。	認証強度については、アルゴリズムの強度以上の差はない。ただし、ワンタイムパスワードは認証のみにしか利用できないが、デジタル証明書では身元情報やポリシーを確認することもできる。
デジタル証明書は厳重に保管しなければならない。	厳重に管理すべきものは秘密鍵であって、デジタル証明書は公開しても問題はない。また、デジタル証明書は、公開することで意味を持つ。
認証は証明書を相手に提示することで行われる。	認証は、公開鍵暗号によるチャレンジ & レスポンスによって、対応する秘密鍵を所持していることで検証される。
証明書の正当性はディレクトリサーバによって保証される。	デジタル証明書の正当性は、CA による署名を検証することで保証される。
CA サーバは認証サーバである。	CA は、単にデジタル証明書に署名して発行するだけであり、認証を実施するのは、デジタル証明書を利用するクライアントとサーバとなる。
CA サーバは常時オンラインで参照できなければならない。	単にデジタル証明書を発行する機能のみであればオンラインである必要はない。ただし、ディレクトリサービスや証明書の申請機能を併用するときにはオンラインである必要がある。

3.4 秘密情報を保存する所

認証のためにクライアントとサーバが共有する「秘密」を使用するパスワード認証の場合、固定パスワード・ワンタイムパスワードでの「秘密」の扱いは表 2 に示すように、ちょうど対称的になります。

表 2：パスワードの存在位置

	ネットワーク上	サーバ上
再利用可能パスワード	パスワードそのもの	パスワードのハッシュ値
ワンタイムパスワード	パスワードのハッシュ値	パスワードそのもの

このように、クライアントとサーバが秘密を共有する方式では、ネットワーク上かサーバ上のいずれかに「秘密」が存在してしまいます。これに対して、公開鍵暗号による認証では、「秘密」（公開鍵暗号の秘密鍵）をネットワーク上で受け渡す必要も、サーバ上で保持する必要もなく、クライアント側でのみ知っていればよいという特徴があります。

いずれにしても、「共有秘密」にせよ、「公開鍵暗号の秘密鍵」にせよ、その情報をクライアントの安全な場所に保管しておく必要があります。現状では、「秘密」をハードディスク上に暗号化して保存し、ローカルなパスワード、PIN（Personal Identification Number）、カード、指紋等によって活性化して利用する方法が多く使われています。また、トークンカード、IC カード、スマートカード等の他の媒体に「秘密」そのものを格納することで、安全性はより一層高まります。

4 リモートアクセス

イントラネットが普及し、外部から社内のシステムにリモートアクセスしたいという要求が増えてきています。さらに、ダイアルアップによる PPP 接続の技術が一般化したことによって、この要求はより一層高まっています。リモートアクセスでは、認証、暗号化、アクセス制御の各技術を適切に組み合わせる必要があります。ただし、リモートアクセスには、次のような問題点もあります。

- バックドアとなりやすく、セキュリティの問題が生じやすい。
- アクセスサーバ、認証サーバ、トークンカード等のリモートアクセス構築や管理のためにコストが発生する。

公衆電話網からのダイアルインによるリモートアクセスで利用できる認証方法は、次の 2 つの部分から成り立ちます。

- 電話網による認証
- PPP における認証

このうち、電話網による認証には、発信者電話番号による方法とコールバックによる方法があります。また、PPP における認証には、平文によるパスワードを利用する PAP と、チャレンジ & レスポンスを利用する CHAP があります。

通常、リモートアクセスでは、ユーザ管理やログ記録のために認証サーバが設置されます。そして、アクセスサーバと認証サーバ間の通信には、デファクトスタンダードとなっている RADIUS が利用されます。ただし、アクセスサーバごとに独自に機能が拡張されていることが多く、設定にはかなり細かい知識が必要になります。また、ワンタイムパスワードを利用する場合、認証サーバが 2 段構成となって、管理が複雑になるケースがあります。

リモートアクセスでのアクセス制御は、クラス分けによる方法が現実的で有効なものとなります。たとえば、一般ユーザはパスワード認証のみでアクセスを許可するがメールサーバにしかアクセスさせず、管理者はトークンカードによって認証し、すべてのネットワーク資源にアクセスできるようにするというような構成は良く見られるものです。

このようなクラス分けを実現するときには、アクセスサーバで接続時に認証を実施し、これを同時にフィルタリングを実施する機器とすることが現時点での現実的な解でしょう。この場合、フィルタリングのルールはアクセスサーバ上で実施し認証サーバ上ではフィルタ番号だけを指定するか、フィルタリングのルールそのものも認証サーバ上で設定することができるアクセスサーバもあります。また、このような考え方は、リモートアクセス VPN でも同様に有効です。

リモートアクセス環境の構築には、独自にアクセスサーバを設置する以外にも、ISP や VAN 企業によるサービスの利用、リモートアクセス VPN の利用という方法がありますが、それぞれのメリット・デメリットを表 3 にまとめてみました。

表 3：リモートアクセスの実現方法の比較

	独自にアクセスサーバを配置	ISP/VPN のサービスを利用	リモートアクセス VPN を利用
発信者電話番号通知	○	△	×
コールバック	○	×	×
ワンタイムパスワード	○	△	△
アクセス制御	○	△	○
アクセスポイントの集約	×	○	○
情報の秘匿化	○	△	○

独自にアクセスサーバを配置すると、その導入や管理のためのコストが問題となります。また、回線や機器の増強やアップグレードにも多大な労力が必要となります。

これに対して、ISP や VAN 企業によるサービスを利用すると、安全性が確保され、管理等の作業もアウトソーシングできます。ただし、認証やアクセス制御等で希望通りの機能が実現できない場合があります。

リモートアクセスに VPN を利用すると、通常のように ISP にダイアルアップ接続し、インターネット経由の VPN によって、社内システムにアクセスすることになります。

4.1 リモートログイン

インターネットを経由したリモートログインでは、パケット盗聴によって平文パスワードや情報が詐取されてしまう可能性があるため、安全な認証や暗号化が必要です。このためには、次に挙げるようなアプリケーションや技術が利用できます。

- ワンタイムパスワードによる telnet や ftp の利用
- SSL-Telnet
- SSH（Secure Shell）
- PET（Privacy Enhanced Telnet）
- VPN（IPSec、PPTP）によるリモートアクセス

このうち SSH は、リモートログインのための r- コマンドの置き換えで、強力な暗号化と認証の機能を提供してくれます。また、ポートフォワーディング機能によって X Window や POP を SSH 上で安全に利用することができます。ただし、ポートフォワーディングでは VPN のようにすべてのアプリケーションが透過的に利用できるわけではなく、その利用方法も多少面倒です。SSH を普及させるためには、やはり、クライアントとなる Windows 環境への対応が重要となります。現在、いくつかの Windows 用 SSH クライアントが提供されていますが、日本語環境では、端末エミュレータTeraTerm用のSSHプラグインがもっとも適していると思われます。

また、インターネット経由での社内メールサーバへのアクセスに限れば、日常的に大量のメールを処理するのでなければ、Web/Mail ゲートウェイやメールサーバによる Web サービス等の利用も検討に値する方法です。このような利用方法は、特に IMAP サーバと組み合わせたときに有効なものとなります。

5 VPN

IPSec（IP Security）が標準化されたことで、VPN もネットワーク構築における実際的な選択肢の 1 つとなってきています。ただし、VPN 自体がまだ新しい技術であるため、「どのような製品や技術を組み合わせればよいか」という点がわかりづらい点があります。

まず、現在利用できる暗号化技術を適用されるネットワーク階層で分類すると、表 4 のようになります。

表 4：各ネットワーク階層に適用されている暗号化技術

データリンク層	Ethernet や WAN の暗号化装置、PPTP（PPP）
ネットワーク層	IPSec
トランスポート層、セッション層	SSL/TLS、SOCKS V5 の暗号化
アプリケーション層	SSH、SSL-Telnet、PET 等によるリモートログイン、PGP、S/MIME 等による暗号化メール

通常、ネットワーク階層的には、下位層に暗号化技術を適用することで、上位層では透過的にその恩恵を受けることができます。ただし、実際には、目的に応じて、適切に暗号化技術を組み合わせて使用する必要があります。

また、当初 LAN 間の接続が中心であった VPN は、現在では次の 2 つの形態で利用されるようになってきています。

- LAN 間接続 VPN
- リモートアクセス VPN

このうち LAN 間接続 VPN は、図 5 に示すように WAN 接続の置き換えとして利用され、専用線よりもコストを削減できるとされています。ただし、実際には、専用線での帯域や遅延といった特性のすべてを置き換えることは困難です。

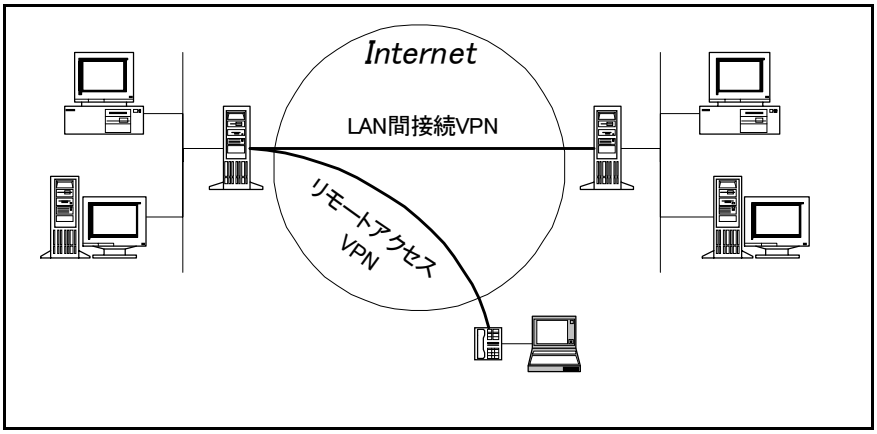

図 5：VPN の利用形態

これに対してリモートアクセス VPN は、アクセスサーバによるダイアルアップ接続の置き換えとして利用され、電話料金やアクセスサーバの管理コストを削減できるものとして期待されています。

このような 2 つの利用形態は、VPN であっても、検討すべき項目や求められる機能はそれぞれ異なったものとなります。VPN ということを忘れてネットワークの接続形態で分類すると、LAN 間接続 VPN は WAN 接続に相当し、リモートアクセス VPN はアクセスサーバに対するダイアルアップ接続に相当します。そして、このように考えると、それぞれの VPN で利用される技術や必要となるノウハウが異なってくることが理解できると思います。

ネットワーク層における VPN を実現するために利用されるセキュリティプロトコルの代表が IPSec です。IPsec は、IPv4 と IPv6 の両方に対して、アドレス・ヘッダ・データの改ざん防止と暗号化の機能を提供します。また、IPSec では、プロトコルの枠組みと、具体的な暗号化や認証の方式、鍵管理方式が分離されているのが特徴です。IPSec の規格は、1995 年 8 月以降に RFC1825 ～ 1829 として規定され、1998 年 11 月以降に RFC2401 ～ RFC2412 として改訂されています。この新しい IPSec の規格では、リプレー攻撃を防止するための Sequence Number Field が新たに設定され、これまで AH（Authentication Header）のみで提供されていたパケット認証の機能が、ESP（Encapsulating Security Payload）でも提供されるようになっています。また、鍵管理プロトコル IKE（Internet Key Exchange）も標準化されています。

IPSec では、データ部分のみを暗号化するトランスポートモードと、IP ヘッダまでも含めて暗号化するトンネルモードの 2 種類の暗号化が利用できます。このうちトランスポートモードは、図 6 のようなホスト間通信での利用に有効なものとなります。

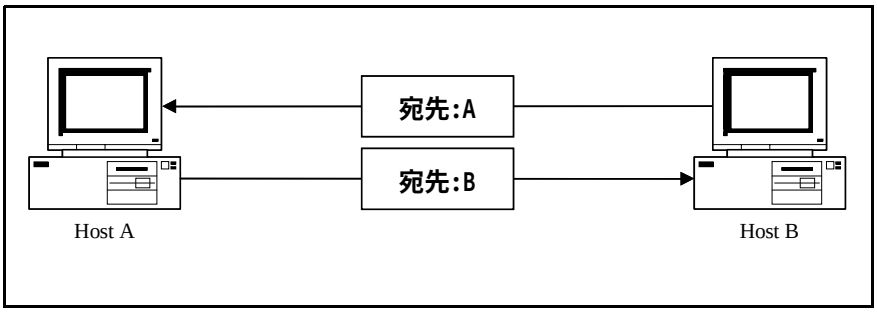

図 6：トランスポートモードでの利用

また、トンネルモードは、図 7 のような VPN での利用に有効なものとなっています。

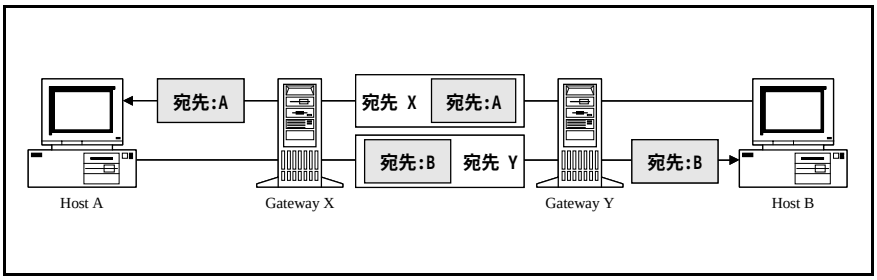

図 7：トンネルモードでの利用

5.1 IPSec での鍵管理

IPSec での鍵管理は、その管理方法から次の 2 種類に分かれます

- 手動鍵管理
- 自動鍵管理

このうち手動鍵管理では、パケット認証や暗号化のパラメータを管理者自らが設定します。これに対して、自動鍵管理では、必要なパラメータが動的に生成され、自動的に設定されます。そして、このときに IKE が鍵管理プロトコルとして標準的に利用されています。

IKE では、次の 2 つの段階に分けて SA（Security Association）が確立されます。

1. IKE 自体が使用する SA を確立する。
2. IPSec で使用する SA を確立する。

また、IPSec では、パケット認証や暗号化のパラメータが動的に決定されるため、それ以前の IKE による接続確立での相互認証が重要なものとなります。現在、IKE の認証方式には、次の 2 種類があります。

- Shared Secret
- Digital Certificate（X.509）

前者は、対向の VPN 機器に共通のパスワード（shared secret）を設定し、これを使った一種のチャレンジ&レスポンスによる認証を行う方法です。後者は、デジタル証明書を使って、公開鍵暗号を用いて VPN 機器間の認証を行う方式です。ただし、現在 RFC になって標準化されているこれらの認証方式は、LAN 間接続 VPN での利用を中心に考えられたものであるため、後述するように、リモートアクセス VPN に対しては必ずしも十分ではありません。

現在 IPSec は、VPN オプションを提供するファイアウォール製品、VPN 専用製品、ルータ等で利用できます。また、PC-UNIX のうち、BSD は KAME、Linux は SWAN によって、IPSec に対応することができます。Windows 環境では、Windows 2000 から IPSec に対応するようです。

マルチベンダ環境での IPSec の相互接続性は、エクストラネットでの利用を考えると必須のものとなります。ただし、現在までに精力的な努力は続けられていますが、十分な状態とはなっていません。手動鍵管理、Shared Secret による IKE、Digitak Certificate による IKE といった複数段階に分かれた課題があり、実験段階で接続できたとしても安定して利用できるまでには至っていません。また、定期的に鍵情報を交換するための rekey 問

題やリブート時の再接続等も、今後解決しなければならない課題となっています。

5.2 VPN とファイアウォール

VPN とファイアウォールは、論理的には独立した存在ですが、実際には組み合わせて利用されることがしばしばあります。その際、ファイアウォールと VPN の接続との関係は、論理的に次の 2 つのパターンが考えられます。

- ファイアウォールの内側同士を接続する（Inside Tunnel）。
- ファイアウォールの外側同士を接続する（Outside Tunnel）。

1 番目のファイアウォールの内側同士を接続する方法では、ファイアウォールによる制限はいっさいなく、任意のアプリケーションが利用できます。このような接続は、同一企業の拠点間を接続するときに利用します。ただし、このときには、VPN のトラフィックがファイアウォールを通過できなければなりません。これに対して 2 番目のファイアウォールの外側同士を接続する方法では、ファイアウォールによる制限を受けるため、ファイアウォールに対応しているアプリケーションのみが利用できます。このような接続は、企業間取引等の他社との接続で利用します。

LAN 間接続 VPN でも、通常の WAN 接続と同様にプライベートアドレスの重複が発生する可能性があります。この問題を一般的に解決する処方箋は存在しないため、ケースバイケースで NAT やプロキシ等の技術を組み合わせて対応するしかありません。ただし、この場合、サービスやアプリケーションの透過性が失われてしまうことがあります。

IPSec による通信は、NAT を経由しては実施できないと基本的に考えるべきです。たとえば、AH でのパケット認証ではアドレス変換は不可能です。新しい IPSec の規格では、ESP ではアドレス変換を可能にする余地があり、NAT による 1 対 1 変換に対応できる可能性があります。ただし、そもそもプロトコルが TCP/UDP ではないので、ポート番号を変換する 1 対多変換の NAT での利用は不可能です。また、IKE では UDP port 500 を使いますが、これもソースポート番号を変換すると動作が保証されません。

5.3 リモートアクセス VPN に必要な機能

前述のように、リモートアクセス VPN では、LAN 間接続 VPN よりも認証が重視されます。これは、リモートアクセス VPN では、不特定の IP アドレスからの接続を受け付けたり、多数のユーザの登録や管理が必要となったりするためです。現在の IPSec/IKE では、Shared Secret 認証の場合、固定 IP アドレスでしか利用できない実装もあります。また、Digital

Certificate は、VPN のためだけでは導入コストや管理コストが大きくなりすぎてしまいます。これらに対して RADIUS サーバとの連携とワンタイムパスワードの利用による方法は、既存システムの継承という意味からも重要であり、現在 IETF によって IPSec での IKE の拡張として検討されているだけでなく、先行して実装された製品も既に提供されています。

また、リモートアクセス VPN では、カプセル化される IP パケットのソースアドレスとして使われるアドレスを割り当てる機能も重要です。このときには、次のようなアドレスの割り当て方が考えられます。

- グローバルアドレスと同じアドレス
- クライアント個別に設定したアドレス
- VPN 装置がアドレスプールから動的に取得したアドレス
- VPN 装置が DHCP を使って取得したアドレス

これらの方法のうち、グローバルアドレスと同じアドレスを割り当てた場合、組織内部とのアドレスとの重複が発生する可能性があります。前 2 者の、静的な割り当てしか利用できない場合は、ネットワーク構成への制約や、アドレスの配布・設定が問題となる可能性があります。後 2 者のアドレスを動的に割り当てる方法は、現在 RFC 化されている IPSec の標準では提供されていませんが、これも IETF によって拡張案が検討されていますし、これを先取りして実装した製品も既に登場しています。

リモートアクセス VPN では、ユーザ別やグループ別のアクセス制御が必要なことがあります。このときの考え方は、前述のアクセスサーバと同様のものとなります。ただし、リモートアクセス VPN では、ユーザ認証の結果に基づいて、VPN 装置が異なるアクセス制御を実施できなければなりません。別法として、ユーザごとやグループごとに固定アドレスを割り当てておき、それを基にファイアウォールやサーバ側でアクセス制御するという方法もあるでしょう。

これまでに示してきた、IPSec でリモートアクセス VPN を実現する際のいくつかの課題に対して、IPSec 以外の解として、たとえば Microsoft 社が提案したプロトコルである PPTP（Point to Point Tunneling Protocol）があります。ただし、PPTP は Microsoft 社の独自方式ですし、過去にセキュリティ専門家からその欠陥が指摘されたりしていることが問題でしょう。

また、PPTP を発展させた L2TP（Layer 2 Tunneling Protocol）は、多くのベンダに支持され実装されていますが、ISP が VPN サービスを提供するために利用するプロトコルとしての位置付けが強いように思えます。ただし、IPSec/IKE を拡張していくよりも、IPSec と L2TP を組み合わせるほうが、リモートアクセス VPN のためには適切であると考えているベンダもあるようです。