Network Working Group
Internet Draft
Expires: April 2003

M. Duffy
Quarry Technologies
October 2002

Framework for IPsec Protected Virtual Links for PPVPNs

draft-duffy-ppvpn-ipsec-vlink-00.txt

Status of this Memo

Abstract

This memo explores some choices that arise when IPsec is to be used to implement secure "virtual links" interconnecting customer premises VPN devices and/or network based virtual routers. The main focus is on the network based cases.

Requirements are proposed and some relevant aspects of the IPsec protocol suite are discussed. A number of different protocol architectures for virtual links are then evaluated.

This memo is informational in nature; it is intended that it will focus discussion toward a standard in this area.

Table of Contents

1. Introduction

A number of different technologies have been identified for implementing Provider Provisioned Virtual Private Networks (PPVPNs). Among the options for providing VPN services at layer 3 are virtual router based VPNs, CE-based IPsec VPNs, and BGP/MPLS VPNs. All three types can capitalize on the creation of IPsec-protected virtual links between VPN-aware nodes at the edge of the network (either PE or CE based). Both the virtual router approach and the dynamically routed flavor of CE-based IPsec approach require a virtual link mechanism, while the reach of BGP/MPLS VPNs can be extended by using virtual links to include remote sites.

Although all these VPN types can use virtual links that carry IP packets across an IP network the VR-based L3 VPN overlay environment [VR-VPN] is the most demanding in terms of a solution. This is because the VR approach requires a virtual link solution that simultaneously supports multiple links for multiple contexts, between a given pair of devices.

A number of drafts have been published dealing with the CE-CE case but there has not been much discussion of the VR (multi-context) case.

This memo assumes some familiarity with the IPsec protocol suite including [IPSEC] and [IKE].

The principal content of this memo is organized as follows: Section 3 lists some desirable properties of a virtual link mechanism. Section 4 describes protocol functions a virtual link system must perform. Section 5 explains some aspects of IPsec that must be considered in using IPsec as a component of a secure virtual link solution. Section 6 presents and evaluates a number of potential protocol architectures.

2. Terminology Used in This Document

CE:	Customer Edge router
PE:	Provider Edge router
Tunnel ingress node:	A system or device that transmits packets into a tunnel
Tunnel egress node:	A system or device that receives packets from a tunnel
Tunnel endpoint:	Either a tunnel ingress node or a tunnel egress node

3. Virtual Link Objectives

For purposes of this memo we define a virtual link as a construct that provides a point-to-point link layer service implemented over an IP network. Each end of a virtual link terminates as an IP interface of a router or virtual router (VR), which may form routing adjacencies and forward IP packets over the link.

The following are viewed as important properties for virtual links:

- Support multiple independent virtual links between a given pair of systems (e.g. PE devices) serving different contexts (e.g. different VR pairs).

- Provide IPsec security services for each virtual link including integrity, data origin authentication, protection against replays, and confidentiality.

- Provide a choice of using IPsec or not, with per-VPN-context and per-remote-PE granularity.

- Support interoperation of network-based (PE-based) virtual routers with Provider Provisioned CE based IPsec VPN devices.

- Operate in systems that simultaneously use IPsec for other purposes.

- Require no new protocol development and minimum extensions to existing protocols.

4. Protocol Functions Needed to Implement Virtual Links

Virtual links are implemented using tunneling technologies. A tunnel, by means of encapsulation, provides isolation for packets sent through it. It thus allows packets to traverse domains they could perhaps not traverse natively, or to be delivered to intermediate destinations not implied by their destination addresses.

Besides encapsulation, tunneling for virtual links requires:

- Multiplexing. The ability to support and distinguish multiple logical streams of data within a tunnel and/or the ability to support and distinguish multiple tunnels between a pair of tunnel endpoints. In fact, the two abilities are logically equivalent and differ only by which entity one applies the term "tunnel" to. In the remainder of this document we shall use the word tunnel in the latter sense i.e. a tunnel carries packets for one VPN context but there may be multiple tunnels between a given pair of tunnel endpoints.

- Tunnel management (setup/maintenance/teardown) signalling. This allows the tunnel endpoints to be explicitly aware of whether a particular tunnel is present or not and perhaps whether it has connectivity (liveliness). In cases where the tunnel mechanism itself does not provide liveliness detection, dynamic routing protocols, if used, can provide this function.

- VPN context correlation. The ability to determine, at the tunnel egress node, what application and context each received packet is intended for. The term "application" here refers to any one of perhaps several functional areas in the node that use tunnels, e.g. virtual link, IPsec security gateway, etc. The "context," for the virtual link application, is a particular VPN i.e. as identified by a [VPN-ID]. Correlation may be done packet by packet in a connectionless manner, however this is likely to impose a high overhead cost. An alternative, available with connection-oriented tunneling technologies, is to establish the correlation on a per-tunnel basis at tunnel setup time, binding the context identification

to a more compact multiplexing field that is transmitted on a per-packet basis.

Support for Path MTU Discovery and for Quality of Service (QoS) on virtual links are important but are not considered in this memo. Reliable or in-order delivery are not required.

5. IPsec Considerations

This section explores a number of choices and issues that must be addressed to use IPsec for virtual links. Although these are separate issues they interrelate heavily with one another.

5.1 Tunnel Mode vs. Transport Mode

IPsec may be used in two different encapsulation modes: IPsec tunnel mode provides in-IP tunneling as a package deal with the security protocol. IPsec transport mode does not provide tunneling. In a virtual link application, IPsec transport mode is of necessity used in conjunction with some other in-IP tunneling protocol for an overall solution.

At first glance tunnel mode seems attractive because it appears to be a one stop shop providing encapsulation, multiplexing, and (via IKE) explicit tunnel management. However, there are difficulties with the semantics of the security policy and with VPN context correlation, as described in the following subsections.

Decoupling tunneling and IPsec and providing them independently yields greater modularity, generally recognized as a Good Thing. Keeping tunneling and IPsec separate also allows for selective use of IPsec since the needed virtual link functions of encapsulation, multiplexing, tunnel management, and correlation are provided separately and there is no reliance on IPsec for these services.

5.2 The Security Policy Database (SPD)

IPsec uses the IKE protocol to negotiate Security Associations (SAs) and their keying. A fundamental aspect of this is the Quick Mode negotiation, via Client IDs, of the access control policy (packet selectors) for each IPsec SA. IPsec devices base this negotiation on their provisioned Security Policy Databases (SPDs). A nominally separate SPD exists for each interface on which IPsec is to run.

The access control policy is one of the basic security services provided by IPsec. It is packet classification against this policy that controls which packets are sent on, or must be received on, a

given SA. By contrast, when using virtual links, we generally want to use a routing decision to determine which packets are sent on a given virtual link, and we generally don't require validation that packets were received from a "correct" link. In a sense, we want a service that is like link-level encryption (, authentication, etc.) for the virtual link. Thus, we have a mismatch between the way IPsec works and the way we want virtual links to work. The nature and extent of this mismatch depends in large part of the relationship between IPsec SAs and virtual links: *Is* a tunnel mode SA the virtual link, or does the virtual link have an existence of its own to which transport mode IPsec may be applied?

5.2.1 Tunnel Mode IPsec and the SPD

If IPsec tunnel mode is used to implement a virtual link we would expect the negotiated selectors to apply to the headers of the "inner packets" e.g. the packets of the VPN. Generally, these packets will be assigned to virtual links based on dynamic routing state and therefore the set of packets to be sent over a particular virtual link are not known a priori. From the point of view of IPsec policy, this is essentially an arbitrary set of packets. What is needed then is an SA that will carry any packets -- an SA whose selectors are all wildcards. However, when multiple virtual links are required between the same pair of tunnel endpoints (e.g. for multiple VPN contexts) multiple SAs with the same wildcard client IDs must be negotiated. Classic IPsec will not generally negotiate multiple SAs out of a given SPD, those SAs having the same client IDs, since in the classic case it has no way to determine which to use for a given outgoing packet. Resolving this requires a slightly liberal interpretation of IPsec, to have an SPD per virtual interface. Unfortunately, it then becomes problematic for the IKE responder to determine which SPD to evaluate an incoming proposal against; this is discussed in section 5.3.

5.2.2 Transport Mode IPsec and the SPD

If another protocol is used to implement the tunnels, with IPsec applied in transport mode, we have essentially positioned the routing and tunneling a layer above IPsec. In this case then the negotiated IPsec selectors might reasonably be expected to apply to the packet headers of the "outer packets" of the tunnel e.g. the GRE, L2TP, IP-in-IP, etc. packets. The selectors match the endpoint addresses of the tunnel and the tunnel protocol. There is no need to create multiple SAs with the same client IDs, and no need for an SPD per virtual interface.

This would seem to be a better match with the access control functions of IPsec. However, if IPsec is controlled solely by

evaluating SPD policy against already-encapsulated packets, it cannot provide much in the way of differentiated protection. In particular, if the tunneling mechanism used is such that all tunnels between a pair of systems have the same outer addresses, protocol, and ports, then we cannot use the SPD to meet the goal of selecting IPsec on a per-VPN basis.

5.2.3 A Hybrid Transport Mode Approach

Another possibility is to use a separate tunneling protocol with transport mode IPsec, but negotiate and apply IPsec policy on the "inner" packets. This opens the possibility for a richer range of differentiated protection than the basic transport mode approach (in which the selectors of packets in the tunnel are invariant). However, this requires a very liberal interpretation of IPsec. As such, it may be difficult to implement in some environments and it may engender strong objections from the IPsec community.

5.3 VPN Context Correlation

For a system that is maintaining multiple contexts (e.g. multiple virtual routers) and which may therefore maintain multiple virtual links to a given remote system, it is essential that there be a way to determine at the downstream end which links are for which contexts. In the cases where IPsec SAs are used to multiplex the virtual links this implies a need to determine which of potentially multiple IPsec applications in the system (e.g. Security Gateway service, IPsec-protected virtual links, etc.) a given SA is for. For those SAs intended for virtual links, it is further necessary to associate them with the correct VPN context.

Several recent Internet-Drafts ([CE-VPN], [TOUCH], [KNIGHT]) have discussed this area but they are all focused on the CE-based VPN case and do not address the multiplexing of multiple contexts between a pair of PE devices. They do address the question of determining whether an SA is for a virtual link or not. These drafts propose adopting the following convention: an IKE proposal for transport mode IPsec for protocol IP-in-IP and client IP addresses that match the IKE endpoint addresses implies that the tunnel will be viewed as a virtual link and routing adjacencies may be formed on it. This convention is expedient, but it is implicit rather than explicit, and it relies on an expectation that existing systems are not already using such proposals under other circumstances. Also, it is not extensible to cover other possible applications.

There are several ways that correlation info (e.g. a VPN-ID) could be passed explicitly from the IKE initiator (which presumably knows it) to the responder (which needs to know it) and bound to a particular

SA they negotiate. Vendor specific payloads could be defined to carry the application identifier (e.g. virtual link) and the context (e.g. VPN-ID) in IKE. However, vendor specific payloads are not a good approach to standardization. New standardized IKE payloads could be defined, but this is unlikely to happen for IKE given the current focus of the IPsec working group on developing its successor. The now-expired [LORDELLO] proposed defining such new payloads within a new ISAKMP Domain of Interpretation (DOI).

It is also possible to pass the correlation info implicitly from initiator to responder, by addressing the IKE proposal to different IP addresses belonging to the responder and/or by presenting different initiator addresses or IKE identities belonging to the initiator. This approach has a number of disadvantages: it is crude, and it requires the maintenance of a tunnel endpoint IP address per VPN context. Even at that it does not convey a VPN identification in absolute terms; rather, coordinated provisioning is still needed at both ends to establish which IP address corresponds with which VPN-ID, etc. Furthermore, the scalability is severely decreased because this forces a separate (and computationally expensive) ISAKMP SA to be needed for each context.

6. Some Possible Protocol Architectures

This section presents six different approaches to constructing virtual links secured by IPsec. Each is evaluated against the requirements and considerations described in the preceding sections.

6.1 Tunnel Mode SA as Virtual Link

This approach uses a tunnel mode IPsec SA to realize each virtual link. Multiple virtual links between a pair of systems, serving different contexts, may be negotiated under a single ISAKMP SA. The client IDs are all-wildcarded.

Each IPsec SA is bound to a VPN-ID through a payload exchanged during the Quick Mode negotiation.

Pros:

. This approach leverages the IKE Quick Mode as a tunnel management protocol.

Cons:

. There is no IPsec-less (unsecured) operation available since it relies on IPsec for tunneling.

- The current IKE standard does not define a payload to convey a VPN-ID; this would require a vendor-specific payload, IKE extension, etc.
- Requires the tunnel end points to support negotiating multiple IPsec SAs with the same client IDs.
- Unlikely to interoperate with CE-based devices.

6.2 Shared IP-in-IP Tunnel with Transport Mode IPsec

This approach uses IP-in-IP tunneling [IP-IP] and a distinct transport mode IPsec SA to realize each virtual link. Multiple virtual links between a pair of systems use the same IP-in-IP tunnel (i.e. the same endpoint addresses). A single ISAKMP SA between a pair of systems is used.

Each virtual link uses a separate transport mode IPsec SA, but they all have the same client IDs. It is the SA (i.e. the SPI field) that distinguishes one virtual link from another. Each IPsec SA is bound to a VPN-ID through a payload exchanged during the Quick Mode negotiation.

Pros:

- This approach leverages the IKE Quick Mode as a tunnel management protocol.
- Likely to interoperate with CE-based devices to form routing adjacencies.

Cons:

- There is no IPsec-less (unsecured) operation available since it relies on IPsec for multiplexing and for tunnel management signaling.
- The current IKE standard does not define a payload to convey a VPN-ID; this would require a vendor-specific payload, IKE extension, etc.
- Requires the tunnel end points to support negotiating multiple IPsec SAs with the same client IDs.

6.3 Distinct IP-in-IP Tunnel with Transport Mode IPsec

This approach uses a distinct IP-in-IP tunnel to realize each virtual link. Multiple virtual links between a pair of systems use distinct IP-in-IP tunnels (i.e. different endpoint addresses). Transport mode IPsec is used to secure the tunnels, and the IKE also provides tunnel management signaling. Each virtual link has its own distinct ISAKMP and IPsec SAs.

Each virtual link terminates on a different tunnel endpoint (i.e. "outer") IP address and it is this that distinguishes one virtual

link from another. The VPN context is bound to each tunnel endpoint address through configuration, directory lookup, or VPN autodiscovery.

Pros:
. This approach leverages the IKE Quick Mode as a tunnel management protocol.
. Likely to interoperate with CE-based devices to form routing adjacencies. This is the proposal of [KNIGHT] and [CE-VPN] extended to multi-context systems.

Cons:
. There is no IPsec-less (unsecured) operation available since it relies on IPsec for the tunnel management signaling. (Unless one does not care about the tunnel management.)
. Requires recognizing, and terminating tunnels on, multiple IP addresses (one per VPN context).
. Scales poorly because it requires an expensive ISAKMP SA per virtual link.
. VPN context correlation requires an external means to associate tunnel endpoint addresses to VPN-IDs and make the association known at both tunnel endpoints.

6.4 GRE Tunnel with Transport Mode IPsec

This approach uses [GRE] to provide the tunneling. Using the Key extension to GRE ([GRE-KEY]) multiple virtual links between a pair of systems are multiplexed within one GRE tunnel. Transport mode IPsec is used when desired to secure the GRE tunnel -- one ISAKMP and one IPsec SA are used per PE pair. Application of IPsec is based on an SPD rule matching the GRE (i.e. "outer") packet header.

A VPN context is bound to each Key value through configuration, directory lookup, or VPN autodiscovery.

Pros:
. IPsec-less operation is available since the tunneling is provided completely independently of IPsec.
. Requires neither extension nor liberal interpretation of IPsec.

Cons:
. Unlikely to interoperate with CE-based devices.
. No tunnel management protocol.
. VPN context correlation requires an external means to associate GRE Key values to VPN-IDs and make the association known at both tunnel endpoints.
. Controlling IPsec use on a per-VPN basis cannot be done within the standard IPsec SPD model, since packets of different VPNs are not

discernible by the selectors available to IPsec (the outer GRE packet).

6.5 MPLS Tunnel with Transport Mode IPsec

This approach uses MPLS-in-IP ([MPLS], [MPLS-IP]) to provide the tunneling. Using an MPLS label in an IP encapsulation, multiple virtual links between a pair of systems are multiplexed within one MPLS-in-IP tunnel. Transport mode IPsec is used when desired to secure the MPLS-in-IP tunnel -- one ISAKMP and one IPsec SA are used per PE pair. Application of IPsec is based on an SPD rule matching the MPLS-in-IP (i.e. "outer") packet header.

A VPN context is bound to each MPLS label value. MPLS labels might be distributed by a label distribution protocol, or by configuration, directory lookup, or piggybacked on autodiscovery. If a label distribution protocol is used, that might serve as a tunnel management protocol.

Pros:

. IPsec-less operation is available since the tunneling is provided completely independently of IPsec.
. Requires neither extension nor liberal interpretation of IPsec.

Cons:

. Unlikely to interoperate with CE-based devices.
. VPN context correlation requires an external means to associate MPLS labels to VPN-IDs and make the association known at both tunnel endpoints.
. Controlling IPsec use on a per-VPN basis cannot be done within the standard IPsec SPD model, since packets of different VPNs are not discernible by the selectors available to IPsec (the outer MPLS-in-IP packet).

6.6 L2TP with Transport Mode IPsec

This approach uses [L2TP] to provide the tunneling. Using L2TP sessions carrying PPP sessions, multiple virtual links between a pair of systems are multiplexed within one L2TP tunnel. Transport mode IPsec is used when desired to secure the tunnel -- one ISAKMP and one IPsec SA are used per PE pair. Application of IPsec is based on an SPD rule matching the L2TP (i.e. "outer") packet header.

A VPN context is bound to each L2TP session via the exchange of L2TP AVPs (e.g. the End Identifier AVP of L2TPv3 -- a similar AVP could be defined for L2TPv2).

Pros:

- IPsec-less operation is available since the tunneling is provided completely independently of IPsec.
- Requires neither extension nor liberal interpretation of IPsec.
- The L2TP control protocol serves as a tunnel management protocol.
- Other helpful PPP and L2TP features are available such as address negotiation, keepalives, etc.

Cons:

- Unlikely to interoperate with CE-based devices.
- Controlling IPsec use on a per-VPN basis cannot be done within the standard IPsec SPD model, since packets of different VPNs are not discernible by the selectors available to IPsec (the outer L2TP packet).

7. Recommendation

The PPVPN working group should develop a standard for IPsec-protected virtual links for the PE-PE environment and one for the CE-CE environment. If those standards can be one and the same or if the CE-CE one can be a subset of the other it would be a plus.

Of the approaches advanced in this memo, the L2TP based approach (section 6.6) appears to provide the nicest characteristics for the PE-PE case. However, vendors of CPE equipment are unlikely to embrace this approach for the CE-CE case. Also, it is arguably a bit on the heavyweight side.

The distinct IP-in-IP tunnel approach (section 6.3) is also promising as it is likely to interoperate with CE-CE VPN devices, and it does not require extensions to IKE to signal the VPN-ID.

8. Security Considerations

This memo deals with ways in which IPsec may be used to secure virtual links in virtual router based PPVPNs. As such security issues are discussed throughout.

Because the virtual router approach exchanges routing messages in-band with the VPN data on the virtual links, securing those links simultaneously secures both the VPN data plane and control plane (or more accurately, the reachability distribution part of the control plane).

Security beyond the boundaries of the provider-provisioned network is beyond the scope of this memo and indeed, beyond the scope of the solutions described here.

9. References

9.1 Normative References

9.2 Informative References

[CE-VPN] De Clercq, Paridaens, Krywaniuk, and Wang, "An Architecture for Provider Provisioned CE-based Virtual Private Networks using IPsec," (Work in progress) draft-ietf-ppvpn-ce-based-02.txt, June 2002.

[GRE] Farinacci, Li, Hanks, Meyer, and Traina, "Generic Routing Encapsulation (GRE)," RFC 2784, March 2000.

[GRE-KEY] Dommety, G., "Key and Sequence Number Extensions to GRE," RFC 2890, September 2000.

[IKE] Harkins, D. and Carrel, D., "The Internet Key Exchange (IKE)," RFC 2409, November 1998.

[IP-IP] Perkins, C., "IP Encapsulation within IP," RFC 2003, October 1996.

[IPSEC] Kent, S. and Atkinson, R., "Security Architecture for the Internet Protocol," RFC 2401, November 1998.

[KNIGHT] Knight, P. and Gleeson, B., "A Method to Provide Dynamic Routing in IPsec VPNs," (Work in progress) draft-knight-ppvpn-ipsec-dynroute-01.txt, July 2002.

[L2TP] Townsley, Valencia, Rubens, Pall, Zorn, and Palter, "Layer Two Tunneling Protocol L2TP," RFC 2661, August 1999.

[MPLS] Rosen, E., Viswanathan, A., and Callon, R., "Multiprotocol Label Switching Architecture," RFC 3031, January 2001.

[MPLS-IP] Wooster, T., Rekhter, Y., and Rosen, E., "Encapsulating MPLS in IP or GRE," (Work in progress) draft-rosen-mpls-in-ip-or-gre-00.txt, August 2002.

[TOUCH] Touch, J. and Eggert, L., "Use of IPsec Transport Mode for Dynamic Routing,". (Work in progress) draft-touch-ipsec-vpn-04.txt, June 2002.

[VPN-ID] Fox, B. and Gleeson, B., "Virtual Private Networks Identifier," RFC 2685, September 1999.

[VR-VPN] Knight, P. et al, "Network based IP VPN Architecture using Virtual Routers," (Work in progress) draft-ietf-ppvpn-vpn-vr-03.txt, July 2002.

10. Summary for Sub-IP Related Internet Drafts

RELATED DOCUMENTS

The References section lists a number of related documents. [CE-VPN] and [KNIGHT] in particular discuss IPsec-protected virtual links, however the solution they propose is aimed at CE-based VPNs and is inadequate for PE-based VPNs, serving multiple contexts.

WHERE DOES IT FIT IN THE PICTURE OF THE SUB-IP WORK?

This I-D is intended for the PPVPN Working Group.

WHY IS IT TARGETED AT THIS WG(s)?

This document addresses a component that is essential for Virtual Router and CE-based provider provisioned VPNs, which are within the purview of the PPVPN working group.

JUSTIFICATION

The PPVPN working group has the charter, among other things, to develop virtual router based VPN standards and to provide for security and privacy of user data in a VPN environment.

Author's Addresses

Mark Duffy
Quarry Technologies
8 New England Executive Park
Burlington MA 01803 USA
Email: mduffy@quarrytech.com

Full Copyright Statement