Network Working Group
Internet-Draft
Expires: August 26, 2004

J. Touch
ISI
L. Eggert
NEC
Y. Wang
ISI

February 26, 2004

# Use of IPsec Transport Mode for Dynamic Routing
# draft-touch-ipsec-vpn-07

## Status of this Memo



## Copyright Notice



## Abstract

This document addresses the use of IPsec to secure the links of a multihop network, to secure communication between trusted components, such as may be used for a secure virtual network (VN), overlay, or virtual private network (VPN). It describes how virtual links established by IPsec tunnel mode can conflict with routing and forwarding inside the VN, due to the IP routing dependence on references to interfaces and next-hop IP addresses. It is exactly because IPsec specification is ambiguous on this issue that compliant

implementations cannot be relied upon to avoid conflicts between routing and forwarding inside a VN.

This document proposes a solution, called IIPtran, in which IPIP encapsulation separate from IPsec is used together with transport-mode IPsec. IPIP tunnel encapsulation occurs as a separate initial step, based on a forwarding lookup of the VN packet. After the forwarding lookup, IPsec transport mode processes the resulting (tunneled) IP packet with an SA determined through a security association database (SAD) match on the tunnel header.

IIPtran supports dynamic routing inside the VN without changes to the current IPsec architecture, by establishing a complete virtual topology with IPIP tunnels that supports static and dynamic routing, and then securing it with IPsec transport mode. IIPtran is an interpretation of how to configure any IPsec specification compliant implementations to avoid the aforementioned conflicts.

The document also discusses how IIPtran compares to several alternative mechanisms for VN routing, and their respective impact on IPsec, routing, policy enforcement and interactions with the Internet Key Exchange (IKE), among other issues.

Table of Contents

## 1. Introduction

The IP security architecture (IPsec) consists of two modes, transport mode and tunnel mode [1]. Transport mode is allowed between two end hosts only; tunnel mode is required when at least one of the endpoints is a "security gateway" (intermediate system that implements IPsec functionality, e.g. a router.)

IPsec can be used to secure the links of a virtual network (VN), creating a secure VN. In a secure VN, trusted routers inside the network dynamically forward packets in the clear (internally), and exchange the packets on secure tunnels, where paths may traverse multiple tunnels. Contrast this to the conventional 'virtual private network' (VPN), which often assumes that paths tend to traverse one secure tunnel to resources in a secure core. A general secure VN allows this secure core to be distributed, composed of trusted or privately-managed resources anywhere in the network.

This document addresses the use of IPsec to secure the links of a multihop, distributed VN. It describes how virtual links established by IPsec tunnel mode can conflict with routing and forwarding inside the VN, due to the IP routing dependence on references to interfaces and next-hop IP addresses.

This document proposes a solution called IIPtran that separates the step of IP tunnel encapsulation from IPsec processing. The solution combines a subset of the current IPsec architecture with other Internet standards to arrive at an interoperable equivalent that is both simpler and has a modular specification.

Later sections of this document compare IIPtran to other proposals for dynamic routing inside VPNs, focusing on the impact the different proposals have on the overall IPsec architecture, routing protocols, security policy enforcement, and the Internet Key Exchange (IKE) [9][10]. An appendix addresses IP tunnel processing issues in IPsec related to IPIP encapsulation and decapsulation.

This document assumes familiarity with other Internet standards [1][2], notably with terminology and numerous acronyms therein.

## 2. Problem Description

Virtual networks connect subsets of resources of an underlying base network, and present the result as a virtual network layer to upper-layer protocols. Similar to a real network, virtual networks consist of virtual hosts (packet sources and sinks) and virtual routers (packet transits), both of which can have a number of network interfaces, and links, which connect multiple network interfaces together. Virtual links (also called tunnels, esp. when point-to-point) are one-hop links in the VN topology, but are either direct links or paths (sequences of connected links) in the underlying base network.

Base network hosts and routers can be part of multiple virtual networks at the same time, and their role in the base network does not need to coincide with their role in a virtual network (i.e. base network hosts may act as VN routers or hosts, as may base network routers).

It is important to note that this definition of a VN is more general than some other definitions, where the VN participation of end systems is limited. Some proposals only allow end systems to be part of a single VN, or even only allow them to be part of the VN and not the base network, substituting the VN for the Internet. The definition above explicitly allows hosts and routers to participate in multiple, parallel VNs, and allows layered VNs (VN inside VN).

It can be useful for a VN to secure its virtual links [3][4], resulting in a VPN. This is not equivalent to end-to-end security, but can be useful when end hosts do not support secure communication themselves. It can provide an additional level of hop-by-hop network security to secure routing in the VPN and isolate the traffic of different VPNs.

The topology of an IPsec VPN commonly consists of IPsec tunnel mode virtual links, as required by the IPsec architecture when the communicating peers are gateway pairs, or a host and a gateway [1]. However, this current required use of IPsec tunnel mode can be incompatible with dynamic routing [3].

The next section provides a short overview on IPsec transport and tunnel mode processing, as far as it is relevant for the understanding of the problem scenarios that follow. The following sections discuss routing problems in detail, based on a common example.

## 2.1 IPsec Overview

There are two modes of IPsec, transport mode and tunnel mode [1]. Transport mode secures portions of the existing IP header and the payload data of the packet, and inserts an IPsec header between the IP header and the payload; tunnel mode adds an additional IP header before performing similar operations. This section gives a short overview of the relevant processing steps for both modes.

In transport mode, IPsec inserts a security protocol header into outgoing IP packets between the original IP header and the packet payload (Figure 1) [5][6][11][12]. The contents of the IPsec header are based on the result of a "security association" (SA) lookup that uses the contents of the original packet header (Figure 1, arrow) as well as its payload (esp. transport layer headers) to locate an SA in the security association database (SAD).

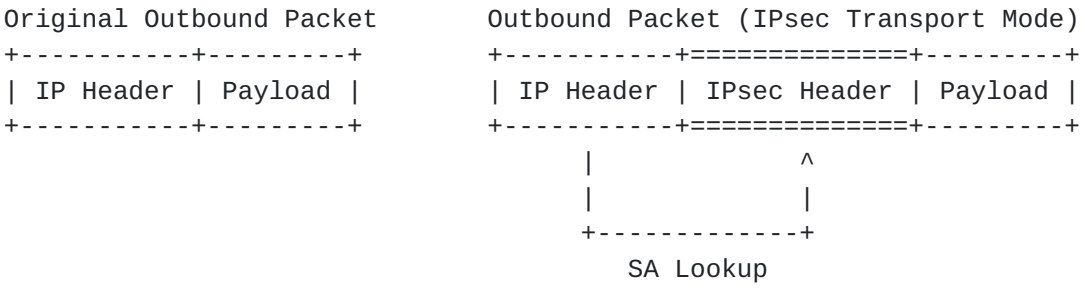


Figure 1: Outbound Packet Construction under IPsec Transport Mode

When receiving packets secured with IPsec transport mode, a similar SA lookup occurs based on the IP and IPsec headers, followed by a verification step after IPsec processing that checks the contents of the packet and its payload against the respective SA. The verification step is similar to firewall processing.

When using tunnel mode, IPsec prepends an IPsec header and an additional IP header to the outgoing IP packet (Figure 2). In essence, the original packet becomes the payload of another IP packet, which IPsec then secures. This has been described [1] as "a tunnel mode SA is essentially a [transport mode] SA applied to an IP tunnel." However, there are significant differences between the two, as described in the remainder of this section.

In IPsec tunnel mode, the IP header of the outbound original packet together with its payload (esp. transport headers) determines the IPsec SA, as for transport mode. However, a tunnel mode SA also contains encapsulation information, including the source and destination IP addresses for the outer tunnel IP header, which is also based on the original outbound packet header and its payload

(Figure 2, arrows).

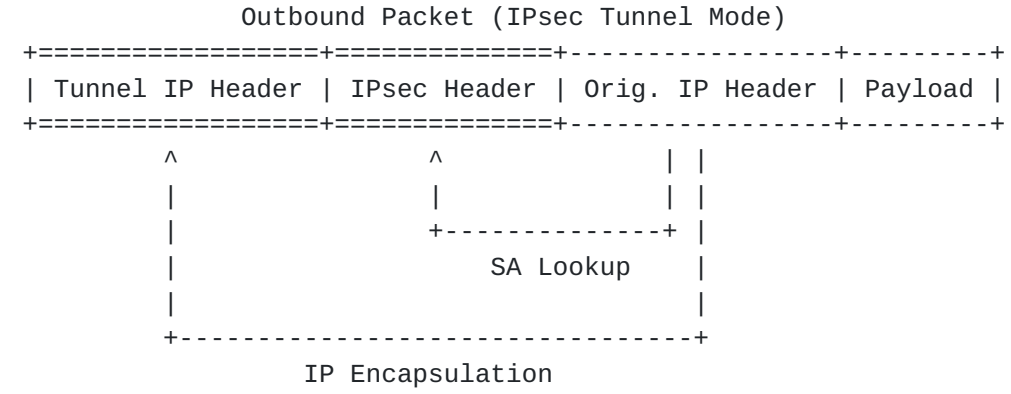


Figure 2: Outbound Packet Construction under IPsec Tunnel Mode

When receiving packets secured with tunnel mode IPsec, an SA lookup occurs based on the contents of the IPsec header and the outer IP header. Next, the packet is decrypted or authenticated based on its IPsec header and the SA, followed by a verification step that checks the contents of the original packet and its payload (esp. the inner IP header and transport headers) against the respective SA.

## 2.2 Forwarding Example

Consider a VPN topology with virtual links established by IPsec tunnel mode SAs, as would be required for compliance with [1]. Such hop-by-hop security can be useful, for example, to secure VN routing, and when legacy end systems do not support end-to-end IPsec themselves.

Virtual routers in a VN need to forward packets the same way regular Internet routers do: based on the destination IP address and the forwarding table. These two determine the next hop IP address the packet should be forwarded to (additional header fields and inner headers can be used, e.g. in policy routing.)

In Figure 3, traffic arrives at gateway A on virtual link 1, having come from any of the virtual hosts upstream of that virtual link. There are two outgoing virtual links for this incoming traffic: out link 3 going to the VPN next-hop gateway B, and out link 4 going to the VPN next-hop gateway C.

For this example, assume the incoming traffic is from a single VPN source X, going to a single VPN destination Y. Ellipses (...) represent multiple virtual links in Figure 3.

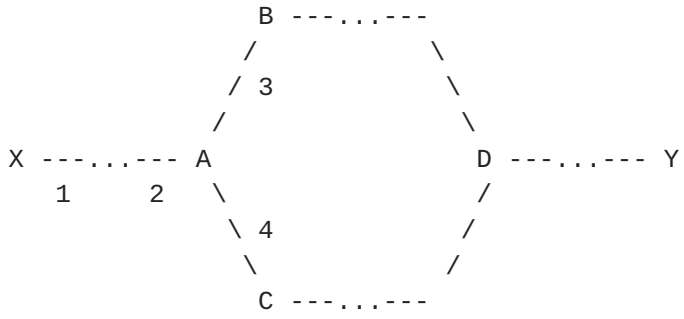


Figure 3: Topology of a Virtual Network

Two problems arise; one is forwarding of VN traffic over IPsec tunnel mode links, the other is source address selection on VN end systems.

## 2.3 Problem 1: Forwarding Issues

Assume a packet from source X to destination Y arrives on link 2 at gateway A. Gateway A now needs to both forward and encrypt the packet to make progress to the next hop gateway inside the VPN.

Dynamically routed gateways forward packets based on a forwarding table managed by a routing daemon that exchanges connectivity information with directly connected peers by communicating on its local interfaces. Entries in the forwarding table map destination IP addresses to the IP address of a next-hop gateway and an associated outbound interface.

The problem is that an intermediate router needs to pick a next hop gateway for a transit packet based on its destination IP address and the contents of the forwarding table. However, the IPsec architecture does not define if and how tunnel mode SAs are represented in the forwarding table.

The problem occurs when A tries to decide how to forward the packet X->Y. In a regular IP network, this decision depends on a forwarding lookup on destination address Y, which indicates the IP address of the next-hop gateway and an associated outbound interface. In the case of a VN, forwarding lookups occur on virtual destination addresses. For the forwarding lookup on such a virtual destination address to succeed, routes through virtual interfaces (tunnels) must exist in the forwarding table.

There are two common implementation scenarios for tunnel mode SAs: One is based on firewall-like packet matching operations where tunnel mode SAs are not virtual interfaces, another is tunnel-based, and treats a tunnel mode SA as a virtual interface. The current IPsec architecture does not mandate one or the other.

Under the first approach, the presence of IPsec tunnel mode SAs is invisible to the IP forwarding mechanism. The lookup uses matching rules in the SA lookup process, closer to firewall matching than traditional IP forwarding lookups, and independent from existing IP forwarding tables. The SA lookup determines which virtual link the packet will be forwarded over, because the tunnel mode SA includes encapsulation information. This lookup and the subsequent tunnel mode processing both ignore the contents of the existing IP forwarding table, whether static or dynamic routing are used. This type of tunnel mode processing is thus incompatible with dynamically routed VPNs.

The second approach - requiring tunnel mode SAs to be interfaces - can be compatible with dynamically routed VPNs (see Section 4) depending on how it is implemented; however, IIPtran (see Section 3) has the additional benefit of greatly simplifying the IPsec architecture and related specifications, and of being compatible with all IPsec specification compliant implementations.

### 2.4 Problem 2: Source Address Selection

A second issue is source address selection at the source host. When an application sends traffic to another host, the host must choose an IP source address for the IP packets before transmission.

When an end system is connected to multiple networks, it must set the source address properly to receive return traffic over the correct network. When a node participates in a virtual network, it is always connected to two networks, the base network and the VN (more if it connects to at least two VNs.) The IPsec specification currently does not define how tunnel mode SAs integrate with source address selection.

For example, when communication occurs over a virtual network, the source address must lie inside the VN. When X sends to Y (Figure 3), the source address must be the IP address of X's local end of tunnel 1. If host A, which has multiple interfaces inside the VN, sends to Y, the source address must be the IP address of the local end of either tunnel 3 or 4.

Most applications do not bind to a specific source IP address, and instead let the host pick one for their traffic [7]. Rules for source address selection that depend heavily on the notions of interfaces and routes.

According to [7], the IP source address of an outbound packet should: (1) for directly connected networks derive from the corresponding interface, or (2) derive from existing dynamic or static route

entries to the destination, or finally (3) derive from the interface attached to a default gateway.

Because IPsec tunnel mode SAs are not required to be interfaces, rules (1) and (2) may not return a usable source address for a given packet. Consequently, VN packets will use the IP address of the local interface connecting to a default gateway as their source address. Often, a default gateway for a host provides connectivity in the base network underlying the VN. The outgoing packet will thus have a source address in the base network, and a destination address in the VN.

This can result in numerous problems, including applications that fail to operation at all, as well as firewalls and admission control failures, and may even lead to compromised security. Consider two cases, one with IPsec tunnels configured with no wildcard tunnel addresses, the other using certain wildcards. In both cases, an application whose source address is set by RFC 1122 [7]rules may send packets (e.g.) with the source address of that host's base network (via the default route) and a destination address of the remote tunnel endpoint.

## 3. IIPtran: IPIP Tunnel Devices + IPsec Transport Mode

This section introduces a solution - called IIPtran - for the two issues identified above. IIPtran replaces IPsec tunnel mode with a combination of IPIP tunnel interfaces that support forwarding and source address selection (as per RFC 2003 [2]), followed by IPsec transport mode on the encapsulated packet.

The IPsec architecture [1] defines the appropriate use of IPsec transport mode and IPsec tunnel mode (host-to-host communication for the former, and all transit communication for the latter). IIPtran appears to violate this requirement, because it uses IPsec transport mode for transit communication.

However, for an IPIP tunnel between security gateways, the gateways themselves source or sink base network traffic when tunneling - they act as hosts in the base network. Thus, IPsec transport mode is also appropriate, if not required, for encapsulated traffic, according to [1].

As a result, replacing IPsec tunnel mode with IPIP tunnel devices and IPsec transport mode is consistent with the existing architecture. Furthermore, this does not compromise the end-to-end use of IPsec, either inside a VPN or in the base network; it only adds IPsec protection to secure virtual links.

The next sections will give a short overview of IPIP encapsulation, and show it combines with IPsec transport mode processing. These section will then discuss how IIPtran addresses each of the problems identified above.

### 3.1 IIPtran Details

IIPtran uses IPIP tunnels (as defined in RFC 2003 [2]), followed by IPsec transport mode on the encapsulated packet.

RFC 2003 [2] uniquely specifies IPIP encapsulation (placing an IP packet as payload inside another IP packet.) Originally developed for MobileIP, it has since often been adopted when virtual topologies were required. Examples include virtual (overlay) networks to support emerging protocols such as IP Multicast, IPv6, and Mobile IP itself, as well as systems that provide private networks over the Internet (X-Bone [3] and PPVPN).

IPIP outbound packet processing, as specified by RFC 2003, [2]tunnels an existing IP packet by prepending it with another IP header (Figure 4.)

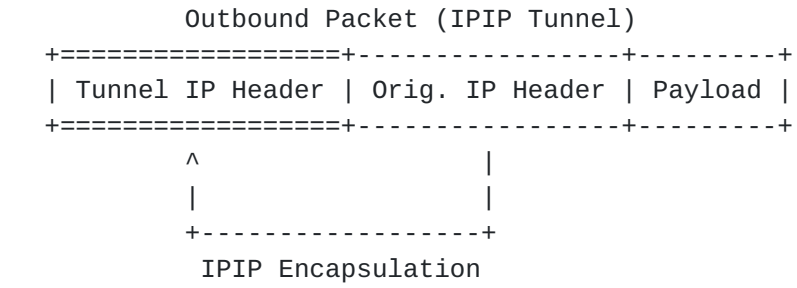


Figure 4: Outbound Packet Construction for IPIP Tunnel

IIPtran performs this IPIP processing as a first step, followed by IPsec transport mode processing on the resulting IPIP packet (Figure 5.)

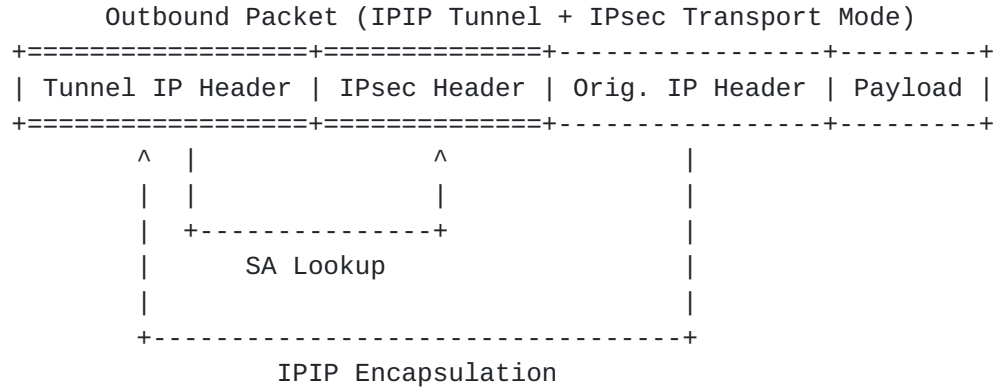


Figure 5: Outbound Packet Construction for IPIP Tunnel with IPsec Transport Mode

A key difference between Figure 2 and Figure 5 is that in the proposed solution, the IPsec header is based on the outer IP header, whereas under IPsec tunnel mode processing, the IPsec header depends on the contents of the inner IP header and payload (see Section 2.1).

However, the resulting VPN packet (Figure 5) on the wire cannot be distinguished from a VPN packet generated by IPsec tunnel mode processing (Figure 2); and the two methods inter-operate, given appropriate configurations on both ends [3].

A detailed discussion of the differences between IIPtran, IPsec tunnel mode, and other proposed mechanisms follows in Section 4. The remainder of this section will describe how IIPtran combines IPIP tunnel devices with IPsec transport mode to solve the problems identified in Section 2.

### 3.2 Solving Problem 1: Forwarding Issues

Section 2.3 described how IP forwarding over IPsec tunnel mode SAs breaks, because tunnel mode SAs are not required to be network interfaces. IIPtran uses RFC 2003 IPIP tunnels [2] to establish the topology of the virtual network. RFC 2003 [2] requires that IPIP tunnels can be routed to, and have configurable addresses. Thus, they can be references in node's routing table (supporting static routing), as well as used by dynamic routing daemons for local communication of reachability information.

RFC 2003 [2] addressed the issue of inserting an IPsec header between the two IP headers that are a result of IPIP encapsulation. IIPtran provides further details on this configuration, and demonstrates how it enables dynamic routing in a virtual network.

It is important to note that the RFC 2003 IPIP tunnels [2] already provide a complete virtual network that an support static or dynamic routing. The proposed solution of using IPIP tunnel with IPsec transport mode decouples IPsec processing from routing and forwarding. IIPtran's use of IPsec is limited to securing the links of the VN (creating a VPN), because IPsec (rightly) lacks internal support for routing and forwarding.

### 3.3 Solving Problem 2: Source Address Selection

Section 2.4 gave an overview of IP source address selection and its dependence on interfaces and routes.

Using RFC 2003 IPIP tunnel devices [2] for VN links, instead of IPsec tunnel mode SAs, allows existing multihoming solutions for source address selection [1] to solve source address selection in this context as well. As indicated in Section 2.4, according to [1], the IP source address of an outbound packet is determined by the outbound interface, which is in turn determined by existing forwarding mechanism. Because IPIP tunnels are full-fledged interfaces with associated routes (as in Section 3.2 of [2]), the routes and address selection as specified in [1] can also operate as desired in the context of VN links.

## 4. Comparison

The previous sections described problems when IPsec tunnel mode provides VPN links, and proposed a solution. This section introduces a number of proposed alternatives, and compares their effect on the IPsec architecture, routing, and policy enforcement, among others, to IIPtran.

### 4.1 Other Proposed Solutions

This section gives a brief overview of a number of alternative proposals that aim at establishing support for dynamic routing for IPsec-secured VNs. The following section then compares these proposals in detail.

Although some of the alternatives also address the issues identified above, IIPtran alone also significantly simplifies and modularizes the IPsec architecture.

#### 4.1.1 Alternative 1: IPsec with Interface SAs

In the first alternative, each IPsec tunnel mode SA is required to act as a full-fledged network interface. This SA interface acts as the outbound interface of the virtual destination's forwarding table entry. IPsec dynamically updates the SA interface configuration in response to SAD changes, e.g. caused by IKE negotiation.

This approach supports dynamic routing and existing source address selection rules, but requires extensions to the IPsec architecture that define tunnel mode SA interfaces and their associated management procedures.

It would necessitate recapitulating the definition of the entirety of RFC 2003 IPIP encapsulation [2], including the association of tunnels with interfaces, inside IPsec. This defeats the modular architecture of the Internet, and violates the specification of type 4 IP in IP packets as being uniquely defined by a single Internet standard (it is already standardized by [2]).

This solution also requires augmenting the IPsec specification to mandate an implementation detail, one that may be difficult to resolve with other IPsec designs, notably the BITS (bump-in-the-stack) alternative. Although the current IPsec specification is ambiguous and allows this implementation, an implementation-independent design is preferable.

### 4.1.2 Alternative 2: IPsec with Initial Forwarding Lookup

A second alternative is the addition of an extra forwarding lookup before IPsec tunnel mode processing. This forwarding lookup will return a "virtual interface" identifier that which indicates how to route the packet [13]. Due to a lack of concrete documentation of this alternative at this time, proposed for an update pending to RFC 2401 [1], two variants are presumed possible:

In the first scenario, the extra forwarding lookup indicates the outbound interface of the final encapsulated tunnel mode packet, i.e. usually a physical interface in the base network. The tunnel mode SA lookup following the forwarding lookup will occur in the per-interface SAD associated with the respective virtual interface.

In the second scenario, the extra forwarding lookup returns an outbound tunnel SA interface. This solution seems to be equivalent to the one described above (Section 4.1.1), i.e. all tunnel mode SAs must be interfaces, and is not discussed separately below.

### 4.1.3 Alternative 3: IPsec with Integrated Forwarding

In the third alternative, the routing protocols and forwarding mechanisms are modified to consult both the routing tables and SADs to make forwarding decision. To prevent IPsec processing from interfering with routing, forwarding table lookup must precede SAD lookup.

This approach supports dynamic routing, but requires changes to routing mechanisms such that SAD contents are included in the route exchanges. It is unclear how transport-layer selectors would affect this approach.

## 4.2 Discussion

This section compares the three different alternatives and IIPtran according to a number of evaluation criteria, such as support for VN forwarding, or impact on the IPsec architecture.

### 4.2.1 VN Routing Support and Complexity

This section investigates whether the three alternatives and IIPtran support VN routing, esp. dynamic routing based on existing IP routing protocols.

Both IIPtran (IPIP tunnels + transport mode) and alternative 1 (per-SA interfaces) establish VN links as full-fledged devices that can be referred to in the routing table, as well as used for local

communication by dynamic routing protocols. They both support static and dynamic VN routing.

However, because the current IPsec architecture does not require tunnel mode SAs to behave similarly to interfaces (some implementers chose alternative 1, but it is not mandated by the specification), alternative 1 requires extensions to the current IPsec architecture that define the exact behavior of tunnel mode SAs. The proposed solution does not require any such changes to IPsec, and for tunnels RFC 2003 already specifies those requirements [2]. Furthermore, addition of those requirements would be redundant and potentially conflict with RFC 2003 [2].

Alternative 3 supports dynamic VN routing, but requires modifying routing protocols and forwarding lookup mechanisms to act or synchronize based on SAD entries. This requires substantial changes to routing software and forwarding mechanisms in all participating nodes to interface to the internals of IPsec; this would require revising a large number of current Internet standards. It is also not clear how tunnel mode SAs that specify port selectors would operate under this scheme, since IP routing has no dependence on transport-layer fields.

Alternative 2 does not support dynamic VN routing. The additional forwarding lookup before IPsec processing is irrelevant, because IPsec tunnel mode SAs are not represented as interfaces, and thus invisible to IP routing protocols.

Additionally, the forwarding lookup suggested for alternative 2 is not compatible with a weak ES model described in [1], which requires both an outbound interface indicator as well as the IP address of the next-hop gateway. For example, multiple tunnels can use the same outgoing interface and thus same SAD. The forwarding lookup would return only the interface; lacking the next-hop gateway, the correct SAD entry cannot be determined. Given the next-hop gateway would not help, because the SAD is not indexed by tunnel mode SA encapsulation destination IP address.

Because alternative 2 fails to support VN routing, it will not be discussed in the remainder of this section.

### 4.2.2 Impact on the IPsec Architecture

IIPtran recognizes that encapsulation is already a property of interface processing, and thus relies on IPIP tunnel devices to handle the IPIP encapsulation for VN links. Tunnel mode IPsec thus becomes unnecessary and can potentially be removed from the IPsec architecture, greatly simplifying the specification.

Alternative 1 requires SAs to be represented as full-fledged interfaces, for the purpose of routing. SAD changes must furthermore dynamically update the configuration of these SA interfaces. The IPsec architecture thus needs extensions that define the operation of interfaces and their interactions with the forwarding table and routes.

Additionally, RFC 2401 [1] describes per-interface SADs as a component of IPsec. When tunnel mode SAs themselves act as interfaces, the function of per-interface SADs needs clarification as follows:

First, each tunnel interface SAD must contain exactly one IPsec tunnel mode SA. Transport mode SAs are prohibited, because they would not result in IP encapsulation (the encapsulation header is part of the tunnel mode SA, a transport mode SA would not cause encapsulation), and thus lead to processing loops. Multiple tunnel mode SAs are prohibited, because dynamic routing algorithms construct topology information based on per-interface communication. Merging different virtual links (tunnels) into a single SA interface can cause routing events on one virtual link to apply incorrectly to other links sharing an SA interface.

Second, only the SAD of physical interfaces may contain IPsec transport mode SAs; otherwise, the current issues with VN routing remain unsolved.

In summary, these restrictions result in only SADs of SA interfaces containing tunnel mode SAs, and only SADs of regular interfaces containing transport mode SAs. Thus, tunnel encapsulation essentially becomes a unique property of the interface, and not IPsec.

IIPtran already recognizes this property. Consequently, it uses IPIP tunnels directly, and combines them with transport mode processing. By eliminating the use of tunnel mode, it removes the need for additional constraints on the contents of per-interface SAs.

### 4.2.3 Policy Enforcement and Selectors

On receiving a packet, both IPsec tunnel mode and IIPtran decrypt and/or authenticate the packet with the same techniques. IPsec tunnel mode decapsulates and decrypts the packet in a single step, followed by a policy check of the inner packet and its payload against the respective IPsec tunnel mode SA. IIPtran uses IPsec transport mode to decrypt and verify the incoming packet, then passes the decrypted IPIP packet on to RFC 2003 IPIP processing [2]. At that point, IIPtran can support selector checks on both the header and its payload using firewall mechanisms, similar to IPsec tunnel mode

processing.

The primary difference between the two is that IPsec tunnel mode does not require a separate processing step for validating packets; once IPsec accepts them during the policy check during decapsulation, they are accepted. IIPtran requires additional processing on the decapsulated packets, to validate whether they conform to their respective IPsec policy.

As noted in Section 5.2 of the IPsec architecture document [1], IPsec processing should retain information about what SAs matched a given packet, for subsequent IPsec or firewall processing. To allow for complex accept policies, it should be possible to reconstruct the format of the original packet at the time it first entered a machine based on saved processing context at any time during inbound processing. IIPtran accepts incoming VN packets only if they have arrived over a specific IPIP tunnel that was secured with IPsec transport mode, but as a separate step following IPIP decapsulation.

Note that IPsec tunnel mode and IIPtran are interoperable [3]. Experiments have verified this interoperability, notably because there are no differences in the resulting packets on the wire, given appropriate keys.

#### 4.2.3.1 Selector Expressiveness

When looking up an SA for a given packet, IPsec allows selectors to match on the contents of the IP header and transport headers. IIPtran using existing IPsec cannot support transport header matches, because SA lookup occurs before decapsulation. A small extension to IPsec can address this issue in a modular way.

RFC 2401 [1] explicitly recognizes that the transport layer header may be nested several headers deep inside the packet, and allows a system to (quote) "chain through the packet headers checking the 'Protocol' or 'Next Header' field until it encounters either one it recognizes as a transport protocol, or until it reaches one that isn't on its list of extension headers, or until it encounters an ESP header that renders the transport protocol opaque."

With IIPtran, the SA lookup starts on the outer (tunnel) header, and selectors including port number information must thus traverse the inner IP header (and possibly other headers) before they can match on the transport headers. IIPtran thus requires that IP be a known IPsec "extension header." This recognizes that with IPIP encapsulation, IP VNs use the base IP network as a link layer. Although this small extension to IPsec is not explicitly required, it is already implied.

Recognizing IP as a valid transport layer over IP also allows selectors to match on the contents of the inner ("transport") IP header. Thus, IPsec selectors under IIPtran can express the same set of policies as conventional IPsec tunnel mode.

Note that in both cases, these policy enforcement rules violate layering by looking at information other than the outermost header. This is consistent with IPsec's current use of port-based selectors. The next section discusses that selectors may not be useful for virtual networks.

#### 4.2.3.2 Role of Selectors for VPNs

For secure VN links established via IPsec tunnel mode SAs, the selectors for the inner (VN) source and destination IP addresses often need to be wildcarded to support dynamic routing in a VN. Thus, the limitation described in 4.2.3.1 (without the proposed extension) may not be important in a VN scenario.

Consider a four-node VN with nodes A, B, C and N (Figure 6). Consider the case where N is either a new node joining an existing VPN, or an existing node that had been disconnected and was just rediscovered via dynamic routing.

In this example, A has IPsec tunnel mode SAs to B and C. If the selectors for the virtual source and destination IP addresses for those SAs are not wildcards, the SA needs to be dynamically modified to permit packets from N to pass over the tunnels to B and C. This becomes quickly impractical as VPN sizes grow.

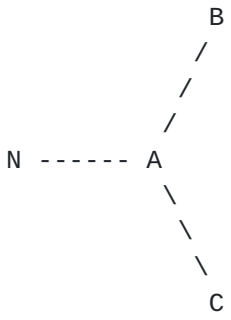


Figure 6: Topology of a Virtual Network

Thus, IPsec selectors appear much less useful in a VPN scenario than expected. A consequence might be that IIPtran - even without extensions to support the full expressiveness of tunnel mode SA selectors as described above - can still support the majority of VPN scenarios.

One purpose of selectors matching on transport header content is policy routing. Different SAs can apply to different applications, resulting in different apparent virtual topologies. IIPtran supports policy routing in a more modular way, by having existing policy routing implementations forward traffic over multiple, parallel VNs. IIPtran supports arbitrary IP-based policy routing schemes, while policies are limited by the expressiveness of IPsec's selectors in the former case.

#### 4.2.4 IKE Impact

The Internet Key Exchange (IKE) [9][10] is a protocol to negotiate IPsec keys between end systems dynamically and securely. It is not a strictly required component of IPsec in the sense that two hosts can communicate using IPsec without having used IKE to negotiate keys (through manually keyed SAs, for example). Despite its name, IKE also acts as a tunnel management protocol (when IPsec tunnel mode SAs are configured), and negotiates security policies between the peers.

Alternatives 1 and 3 use existing IKE without changes.

One possible approach to use IKE with IIPtran is to negotiate a tunnel mode SA, and then treat it as a transport mode SA against an IPIP tunnel when communicating with conventional peers. For policies that do not specify selectors based on transport-layer information, this establishes interoperability.

However, since IIPtran eliminates IPsec tunnel mode, it could also simplify IKE, by limiting it to its original purpose of key exchange. A new tunnel management protocol (e.g. ATMP [8]) would set up IPIP tunnels, use an as of yet unspecified second protocol to negotiate security policy, and then use IKE to exchange keys for use with the policy.

Current IKE operation would become a modular composition of separate protocols, similar to how IIPtran modularizes IPsec by combining existing Internet standards. For example, a VPN link creation could follow these steps: (1) IKE negotiation in the base network to secure (2) a subsequent tunnel management exchange [8] in the base network, followed by (3) IKE exchanges over the established tunnel to create a secure VPN link.

## 5. Security Considerations

This document addresses security considerations throughout, as they are a primary concern of proposed uses of IPsec.

The primary purpose of this document is to extend the use of IPsec to dynamically routed VPNs, which will extend the use of IPsec and, it is hoped, increase the security of VPN infrastructures using existing protocols.

## 6. Summary and Recommendations

This document presents a mechanism consistent with the current use of IPsec which supports dynamic routing inside a virtual network that uses IPsec to secure its links. It illustrates how current use of IPsec tunnel mode can fail to support dynamic VN routing (depending on the implementation), and compares IIPtran with several different alternatives. It finds that IIPtran, a composite of a subset of IPsec (i.e. transport mode) together with existing standard IPIP encapsulation, results in an interoperable, standards-conforming equivalent that is both simpler and modular.

## 7. Acknowledgments

The authors would like to thank the members of the X-Bone and DynaBone projects at USC/ISI for their contributions to the ideas behind this draft, notably (current) Greg Finn and (past) Amy Hughes, Steve Hotz and Anindo Banerjea.

The authors would also like to thank Jun-ichiro (itojun) Hagino and the KAME project for bringing IKE implications of this proposal to our attention, as well as implementing the mechanisms in this draft in the KAME IPv6/IPsec network stack. Members of several IETF WGs (especially IPsec: Stephen Kent, PPVPN: Eric Vyncke, Paul Knight, various members of MobileIP) provided valuable input on the details of IPsec processing in earlier revisions of this document.

Effort sponsored by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreements number F30602-98-1-0200 entitled "X-Bone" and number F30602-01-2-0529 entitled "DynaBone".

Normative References

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

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

[3] Touch, J., "Dynamic Internet overlay deployment and management using the X-Bone", Computer Networks Vol. 36, No. 2-3, July 2001.

[4] Touch, J., Wang, Y., Eggert, L. and G. Finn, "A Virtual Internet Architecture", ISI Technical Report ISI-TR-570, Workshop on Future Directions in Network Architecture (FDNA) 2003, March 2003.

[5] Kent, S. and R. Atkinson, "IP Authentication Header", RFC 2402, November 1998.

[6] Kent, S. and R. Atkinson, "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998.

[7] Braden, R., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, October 1989.

[8] Hamzeh, K., "Ascend Tunnel Management Protocol - ATMP", RFC 2107, February 1997.

## Informative References

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

[10] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", draft-ietf-ipsec-ikev2-12 (work in progress), January 2004.

[11] Kent, S., "IP Authentication Header", draft-ietf-ipsec-rfc2402bis-06 (work in progress), February 2004.

[12] Kent, S., "IP Encapsulating Security Payload (ESP)", draft-ietf-ipsec-esp-v3-07 (work in progress), February 2004.

[13] Kent, S., "Personal Communication", November 2002.

[14] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, November 1990.

[15] Lahey, K., "TCP Problems with Path MTU Discovery", RFC 2923, September 2000.

## Authors' Addresses

Joe Touch
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
US

Phone: +1 310 822 1511
Fax: +1 310 823 6714
EMail: touch@isi.edu
URI: http://www.isi.edu/touch

Lars Eggert
NEC Network Laboratories
Kurfuersten-Anlage 36
Heidelberg 69115
DE

Phone: +49 6221 90511 43
Fax: +49 6221 90511 55
EMail: lars.eggert@netlab.nec.de
URI: http://www.netlab.nec.de/

Yu-Shun Wang
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
US

Phone: +1 310 822 1511
Fax: +1 310 823 6714
EMail: yushunwa@isi.edu
URI: http://www.isi.edu/yushunwa

## Appendix A. Encapsulation/Decapsulation Issues

There are inconsistencies between the IPIP encapsulation rules specified by IPsec [1] and those specified by MobileIP [2]. The latter specification is standards track, and the IP protocol number of 4 (payload of an IP packet of type 4) is uniquely specified by RFC 2003 according to IANA [2]. The use of IPIP inside an IPsec transport packet can be confused with IPsec tunnel mode, because IPsec does not specify any limits on the types of IP packets that transport mode can secure,

### A.1 Encapsulation Issues

When an IP packet is encapsulated as payload inside another IP packet, some of the outer header fields can be newly written (and the inner header determines some others [2].) Among these fields is the IP DF (do not fragment) flag. When the inner packet DF flag is clear, the outer packet may copy it or set it; however, when the inner DF flag is set, the outer header must copy it [2]. IPsec defines conflicting rules, where that flag and other similar fields (TOS, etc.) may be copied, cleared, or set as specified by an SA.

The IPsec specification indicates that such fields must be controlled, to achieve security. Otherwise, such fields could provide a covert channel between the inner packet header and outer packet header. However, RFC 2003 [2] requires that the outer fields not be cleared when the inner ones are set, to prevent MTU discovery "black holes" [14][15].

To avoid a conflict between these rules, and to avoid security weaknesses associated with solely copying the fields, it is recommended that IPsec IPIP encapsulation not permit the clearing of the outer DF flag. When the SA requires clearing the DF flag, and the inner packet DF is set, it is proposed that IPsec drop that packet, rather than violate RFC 2003 processing rules [2]. Similar rules are being developed for TOS and other similar IP header fields, to be included in an update of RFC 2003 [2].

Another approach to closing the covert channel is always to set the DF flag in the outer header (whether or not it is set in the inner header). Setting the DF flag allows PMTU discovery to operate normally. The details of this approach are discussed in [2].

### A.2 Decapsulation Issues

Given identical keys, a packet created by IPIP tunnel encapsulation combined with IPsec transport mode and an IPsec tunnel mode packet look identical on the wire. Thus, when an IPsec'ed packet arrives

that contains an IPIP inner packet, it is not possible to distinguish whether the packet was created using IPsec tunnel mode or IPsec transport mode of an IPIP encapsulated packet. In both cases, the protocol field of the outer header is IPsec (AH or ESP), and the "next header" field for the inner data is 4 (IP). IPsec requires the SA matching a received packet to indicate whether to apply tunnel mode or transport mode.

Incoming packet processing must check the SAD before determining whether to decapsulate IPsec packets with inner payload of protocol type 4. If the SAD indicates that a tunnel mode association applies, IPsec must decapsulate the packet. If the SAD indicates that a transport mode association applies, IPsec must not decapsulate the packet. This requires that the SAD indicate one of these two options; wildcard SAD entries ("ANY", or "TUNNEL or TRANSPORT") cannot be supported.

### A.3 Appendix Summary

IPsec's use of IPIP encapsulation conflicts with the IPIP standard [2], This issue is already being resolved in an update to RFC 2003, instead of specifying a non-standard conforming variant of IPIP encapsulation inside IPsec.

## Intellectual Property Statement



## Full Copyright Statement

## Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.