

Network Working Group
Request for Comments: 3104
Category: Experimental

G. Montenegro
Sun Microsystems, Inc.
M. Borella
CommWorks
October 2001

RSIP Support for End-to-end IPsec

Status of this Memo

This memo defines an Experimental Protocol for the Internet community. It does not specify an Internet standard of any kind. Discussion and suggestions for improvement are requested. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2001). All Rights Reserved.

IESG Note

The IESG notes that the set of documents describing the RSIP technology imply significant host and gateway changes for a complete implementation. In addition, the floating of port numbers can cause problems for some applications, preventing an RSIP-enabled host from interoperating transparently with existing applications in some cases (e.g., IPsec). Finally, there may be significant operational complexities associated with using RSIP. Some of these and other complications are outlined in section 6 of the RFC 3102, as well as in the Appendices of RFC 3104. Accordingly, the costs and benefits of using RSIP should be carefully weighed against other means of relieving address shortage.

Abstract

This document proposes mechanisms that enable Realm Specific IP (RSIP) to handle end-to-end IPsec (IP Security).

Table of Contents

| | |
|--|----|
| 1. Introduction | 2 |
| 2. Model | 2 |
| 3. Implementation Notes | 3 |
| 4. IKE Handling and Demultiplexing | 4 |
| 5. IPsec Handling and Demultiplexing | 5 |
| 6. RSIP Protocol Extensions | 6 |
| 6.1 IKE Support in RSIP | 6 |
| 6.2 IPsec Support in RSIP | 7 |
| 7. IANA Considerations | 10 |
| 8. Security Considerations | 10 |
| 9. Acknowledgements | 10 |
| References | 11 |
| Authors' Addresses | 12 |
| Appendix A: On Optional Port Allocation to RSIP Clients | 13 |
| Appendix B: RSIP Error Numbers for IKE and IPsec Support | 14 |
| Appendix C: Message Type Values for IPsec Support | 14 |
| Appendix D: A Note on Flow Policy Enforcement | 14 |
| Appendix E: Remote Host Rekeying | 14 |
| Appendix F: Example Application Scenarios | 15 |
| Appendix G: Thoughts on Supporting Incoming Connections | 17 |
| Full Copyright Statement | 19 |

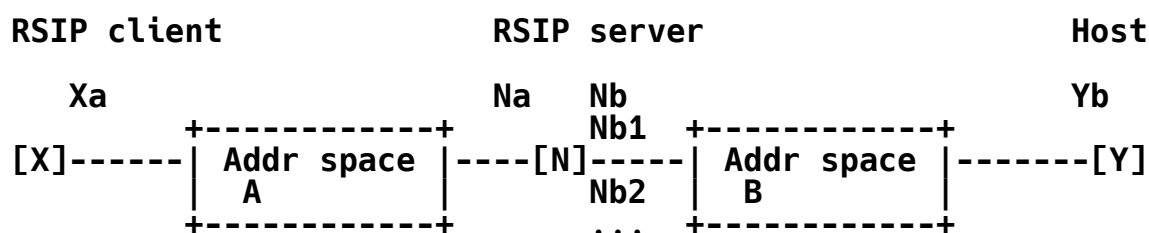
1. Introduction

This document specifies RSIP extensions to enable end-to-end IPsec. It assumes the RSIP framework as presented in [RSIP-FW], and specifies extensions to the RSIP protocol defined in [RSIP-P]. Other terminology follows [NAT-TERMS].

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

2. Model

For clarity, the discussion below assumes this model:



Hosts X and Y belong to different address spaces A and B, respectively, and N is an RSIP server. N has two addresses: Na on address space A, and Nb on address space B. For example, A could be a private address space, and B the public address space of the general Internet. Additionally, N may have a pool of addresses in address space B which it can assign to or lend to X.

This document proposes RSIP extensions and mechanisms to enable an RSIP client X to initiate IKE and IPsec sessions to a legacy IKE and IPsec node Y. In order to do so, X exchanges RSIP protocol messages with the RSIP server N. This document does not yet address IKE/IPsec session initiation from Y to an RSIP client X. For some thoughts on this matter see Appendix G.

The discussion below assumes that the RSIP server N is examining a packet sent by Y, destined for X. This implies that "source" refers to Y and "destination" refers to Y's peer, namely, X's presence at N.

This document assumes the use of the RSAP-IP flavor of RSIP (except that port number assignments are optional), on top of which SPI values are used for demultiplexing. Because of this, more than one RSIP client may share the same global IP address.

3. Implementation Notes

The RSIP server N is not required to have more than one address on address space B. RSIP allows X (and any other hosts on address space A) to reuse Nb. Because of this, Y's SPD SHOULD NOT be configured to support address-based keying. Address-based keying implies that only one RSIP client may, at any given point in time, use address Nb when exchanging IPsec packets with Y. Instead, Y's SPD SHOULD be configured to support session-oriented keying, or user-oriented keying [Kent98c]. In addition to user-oriented keying, other types of identifications within the IKE Identification Payload are equally effective at disambiguating who is the real client behind the single address Nb [Piper98].

Because it cannot rely on address-based keying, RSIP support for IPsec is similar to the application of IPsec for remote access using dynamically assigned addresses. Both cases impose additional requirements which are not met by minimally compliant IPsec implementations [Gupta]:

Note that a minimally-compliant IKE implementation (which only implements Main mode with Pre-shared keys for Phase I authentication) cannot be used on a remote host with a dynamically assigned address. The IKE responder (gateway) needs to look up the initiator's (mobile node's) pre-shared key before it can

decrypt the latter's third main mode message (fifth overall in Phase I). Since the initiator's identity is contained in the encrypted message, only its IP address is available for lookup and must be predictable. Other options, such as Main mode with digital signatures/RSA encryption and Aggressive mode, can accommodate IKE peers with dynamically assigned addresses.

IKE packets are typically carried on UDP port 500 for both source and destination, although the use of ephemeral source ports is not precluded [ISAKMP]. IKE implementations for use with RSIP **SHOULD** employ ephemeral ports, and should handle them as follows [IPSEC-MSG]:

IKE implementations **MUST** support UDP port 500 for both source and destination, but other port numbers are also allowed. If an implementation allows other-than-port-500 for IKE, it sets the value of the port numbers as reported in the ID payload to 0 (meaning "any port"), instead of 500. UDP port numbers (500 or not) are handled by the common "swap src/dst port and reply" method.

It is important to note that IPsec implementations **MUST** be aware of RSIP, at least in some peripheral sense, in order to receive assigned SPIs and perhaps other parameters from an RSIP client. Therefore, bump-in-the-stack (BITS) implementations of IPsec are not expected to work "out of the box" with RSIP.

4. IKE Handling and Demultiplexing

If an RSIP client requires the use of port 500 as its IKE source, this prevents that field being used for demultiplexing. Instead, the "Initiator Cookie" field in the IKE header fields must be used for this purpose. This field is appropriate as it is guaranteed to be present in every IKE exchange (Phase 1 and Phase 2), and is guaranteed to be in the clear (even if subsequent IKE payloads are encrypted). However, it is protected by the Hash payload in IKE [IKE]. Because of this, an RSIP client and server must agree upon a valid value for the Initiator Cookie.

Once X and N arrive at a mutually agreeable value for the Initiator Cookie, X uses it to create an IKE packet and tunnels it the RSIP server N. N decapsulates the IKE packet and sends it on address space B.

The minimum tuple negotiated via RSIP, and used for demultiplexing incoming IKE responses from Y at the RSIP server N, is:

- IKE destination port number
- Initiator Cookie
- Destination IP address

One problem still remains: how does Y know that it is supposed to send packets to X via Nb? Y is not RSIP-aware, but it is definitely IKE-aware. Y sees IKE packets coming from address Nb. To prevent Y from mistakenly deriving the identity of its IKE peer based on the source address of the packets (Nb), X MUST exchange client identifiers with Y:

- IDii, IDir if in Phase 1, and
- IDci, IDcr if in Phase 2.

The proper use of identifiers allows the clear separation between those identities and the source IP address of the packets.

5. IPsec Handling and Demultiplexing

The RSIP client X and server N must arrive at an SPI value to denote the incoming IPsec security association from Y to X. Once N and X make sure that the SPI is unique within both of their SPI spaces, X communicates its value to Y as part of the IPsec security association establishment process, namely, Quick Mode in IKE [IKE] or manual assignment.

This ensures that Y sends IPsec packets (protocols 51 and 50 for AH and ESP, respectively) [Kent98a,Kent98b] to X via address Nb using the negotiated SPI.

IPsec packets from Y destined for X arrive at RSIP server N. They are demultiplexed based on the following minimum tuple of demultiplexing fields:

- protocol (50 or 51)
- SPI
- destination IP address

If N is able to find a matching mapping, it tunnels the packet to X according to the tunneling mode in effect. If N cannot find an appropriate mapping, it MUST discard the packet.

6. RSIP Protocol Extensions

The next two sections specify how the RSIP protocol [RSIP-P] is extended to support both IKE (a UDP application) and the IPsec-defined AH and ESP headers (layered directly over IP with their own protocol numbers).

If a server implements RSIP support for IKE and IPsec as defined in this document, it MAY include the RSIP Method parameter for RSIP with IPsec in the REGISTER_RESPONSE method sent to the client. This method is assigned a value of 3:

3 RSIP with IPsec (RSIPSEC)

Unless otherwise specified, requirements of micro and macro flow-based policy are handled according to [RSIP-P].

6.1 IKE Support in RSIP

As discussed above, if X's IPsec implementation allows use of an ephemeral source port for IKE, then incoming IKE traffic can be demultiplexed by N based on the destination address and port tuple. This is the simplest and most desirable way of supporting IKE, and IPsec implementations that interact with RSIP SHOULD allow it.

However, if X must use source port 500 for IKE, there are two techniques with which X and N can arrive at a mutually unique Initiator Cookie.

- Trial and error.
- Negotiation via an extension of the RSIP protocol.

The trial and error technique consists of X first obtaining resources with which to use IPsec (via ASSIGN_REQUEST_RSIPSEC, defined below), and then randomly choosing an Initiator Cookie and transmitting the first packet to Y. Upon arrival at N, the RSIP server examines the Initiator Cookie for uniqueness per X's assigned address (Nb). If the cookie is unique, N allows the use of this cookie for this and all subsequent packets between X and Y on this RSIP binding. If the cookie is not unique, N drops the packet.

When an IKE packet is determined to be lost, the IKE client will attempt to retransmit at least three times [IKE]. An RSIP-aware IKE client SHOULD use different Initiator Cookies for each of these retransmissions.

The probability of an Initiator Cookie collision at N and subsequent retransmissions by X, is infinitesimal given the 64-bit cookie space. According to the birthday paradox, in a population of 640 million RSIP clients going through the same RSIP server, the chances of a first collision is just 1%. Thus, it is desirable to use the trial and error method over negotiation, for these reasons:

- Simpler implementation requirements
- It is highly unlikely that more than one round trip between X and N will be necessary.

6.2 IPsec Support in RSIP

This section defines the protocol extensions required for RSIP to support AH and ESP. The required message types are ASSIGN_REQUEST_RSIPSEC and ASSIGN_RESPONSE_RSIPSEC:

ASSIGN_REQUEST_RSIPSEC

The ASSIGN_REQUEST_RSIPSEC message is used by an RSIP client to request IPsec parameter assignments. An RSIP client **MUST** request an IP address and SPIs in one message.

If the RSIP client wishes to use IPsec to protect a TCP or UDP application, it **MUST** use the port range parameter (see Appendix A). Otherwise, it **MUST** set the port parameters to the "don't need" value. This is accomplished by setting the length field to 0, and by omitting both the number field and the port field. This informs the server that the client does not actually need any port assignments.

The client may initialize the SPI parameter to the "don't care" value (see below). In this case, it is requesting the server to assign it a valid SPI value to use.

Alternatively, the client may initialize the SPI parameter to a value it considers valid. In this case, it is suggesting that value to the server. Of course, the server may choose to reject that suggestion and return an appropriate error message.

The format of this message is:

```
<ASSIGN_REQUEST_RSIPSEC> ::= <Version>
                                <Message Type>
                                <Overall Length>
                                <Client ID>
                                <Address (local)>
                                <Ports (local)>
                                <Address (remote)>
                                <Ports (remote)>
                                <SPI>
                                [Message Counter]
                                [Lease Time]
                                [Tunnel Type]
```

The following message-specific error conditions exist. The error behavior of ASSIGN_REQUEST_RSIP_IPSEC follows that of ASSIGN_REQUEST_RSAP-IP for all non-IPsec errors.

- If the client is not allowed to use IPsec through the server, the server MUST respond with an ERROR_RESPONSE containing the IPSEC_UNALLOWED parameter.
- If the SPI parameter is a "don't care" value and the RSIP server cannot allocate ANY SPIs, the RSIP server MUST respond with an ERROR_RESPONSE containing the IPSEC_SPI_UNAVAILABLE error.
- If an SPI parameter is not a "don't care" value and the RSIP server cannot allocate it because the requested address and SPI tuple is in use, the RSIP server MUST respond with an ERROR_RESPONSE containing the IPSEC_SPI_INUSE error.

ASSIGN_RESPONSE_RSIPSEC

The ASSIGN_RESPONSE_RSIPSEC message is used by an RSIP server to assign parameters to an IPsec-enabled RSIP client.

The format of this message is:

```
<ASSIGN_RESPONSE_RSIPSEC> ::= <Version>
                                <Message Type>
                                <Overall Length>
                                <Client ID>
                                <Bind ID>
                                <Address (local)>
                                <Ports (local)>
                                <Address (remote)>
                                <Ports (remote)>
                                <SPI>
                                <Lease Time>
                                <Tunnel Type>
                                [Address (tunnel endpoint)]
                                [Message Counter]
```

If the port parameters were set to the "don't need" value in the request (see above), the RSIP server must do the same in the response.

Additionally, RSIP support for IPsec requires the following new parameter:

SPI

| Code | Length | Number | SPI | | SPI |
|---------|---------|---------|---------|-----|---------|
| +-----+ | +-----+ | +-----+ | +-----+ | | +-----+ |
| 22 | 2 | 2 bytes | 4 bytes | ... | 4 bytes |
| +-----+ | +-----+ | +-----+ | +-----+ | | +-----+ |

Sent by the RSIP client in ASSIGN_REQUEST_RSIPSEC messages to ask for a particular number of SPIs to be assigned. Also sent by the RSIP server to the client in ASSIGN_RESPONSE_RSIPSEC messages.

The "SPI" fields encode one or more SPIs. When a single SPI is specified, the value of the number field is 1 and there is one SPI field following the number field. When more than one SPI is specified, the value of the number field will indicate the total number of SPIs contained, and the parameter may take one of two forms. If there is one SPI field, the SPIs specified are considered to be contiguous starting at the SPI number specified in the SPI field. Alternatively, there may be a number of SPI fields equal to the value of the number field. The number of SPI fields can be extrapolated from the value of the length field.

In some cases, it is necessary to specify a "don't care" value for one or more SPIs. This is accomplished by setting the length field to 2 (to account for the 2 bytes in the Number field), setting the number field to the number of SPIs necessary, and omitting all SPI fields. The value of the number field **MUST** be greater than or equal to one.

7. IANA Considerations

All of the designations below are tentative.

- RSIP IPsec error codes (see below).
- ASSIGN_REQUEST_RSIP_IPSEC message type code.
- SPI parameter code.

8. Security Considerations

This document does not add any security issues to those already posed by NAT, or normal routing operations. Current routing decisions typically are based on a tuple with only one element: destination IP address. This document just adds more elements to the tuple.

Furthermore, by allowing an end-to-end mode of operation and by introducing a negotiation phase to address reuse, the mechanisms described here are more secure and less arbitrary than NAT.

A word of caution is in order: SPI values are meant to be semi-random, and, thus serve also as anti-clogging tokens to reduce off-the-path denial-of-service attacks. However, RSIP support for IPsec, renders SPI's a negotiated item: in addition to being unique values at the receiver X, they must also be unique at the RSIP server, N. Limiting the range of the SPI values available to the RSIP clients reduces their entropy slightly.

9. Acknowledgements

Many thanks to Bernard Aboba, Vipul Gupta, Jeffrey Lo, Dan Nessel, Gary Jaszewski and Prakash Iyer for helpful discussions.

References

- [Gupta] Gupta, V., "Secure Remote Access over the Internet using IPsec", Work in Progress.
- [IKE] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.
- [ISAKMP] Maughan, D., Schertler, M., Schneider, M. and J. Turner, "Internet Security Association and Key Management Protocol (ISAKMP)", RFC 2408, November 1998.
- [IPSEC-MSG] Ted Ts'o, message to the IETF's IPsec mailing list, Message-Id:<199911232216.RAA01932@trampoline.thunk.org>, November 23, 1999.
- [Jenkins] Jenkins, T., "IPsec Rekeying Issues", Work in Progress.
- [Kent98a] Kent, S. and R. Atkinson, "IP Encapsulating Payload", RFC 2406, November 1998.
- [Kent98b] Kent, S. and R. Atkinson, "IP Authentication Header", RFC 2402, November 1998.
- [Kent98c] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.
- [Piper98] Piper, D., "The Internet IP Security Domain of Interpretation for ISAKMP", RFC 2407, November 1998.
- [NAPT] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.
- [NAT-TERMS] Srisuresh, P. and M. Holdredge, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.
- [RSIP-FW] Borella, M., Lo, J., Grabelsky, D. and G. Montenegro, "Realm Specific IP: A Framework", RFC 3102, October 2001.
- [RSIP-P] Borella, M., Grabelsky, D., Lo, J. and K. Taniguchi, "Realm Specific IP: Protocol Specification", RFC 3103, October 2001.

Authors' Addresses

Gabriel E. Montenegro
Sun Microsystems
Laboratories, Europe
29, chemin du Vieux Chene
38240 Meylan
FRANCE

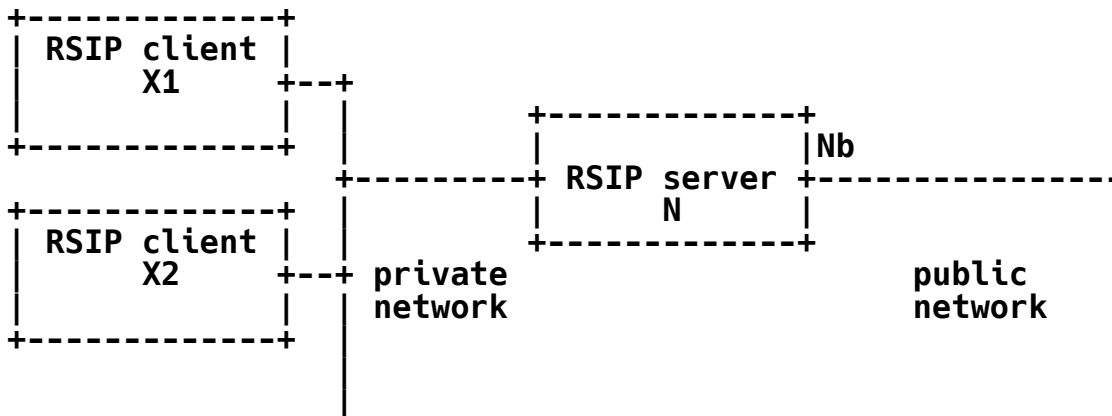
Phone: +33 476 18 80 45
EMail: gab@sun.com

Michael Borella
CommWorks
3800 Golf Rd.
Rolling Meadows IL 60008

Phone: (847) 262-3083
EMail: mike_borella@commworks.com

Appendix A: On Optional Port Allocation to RSIP Clients

Despite the fact that SPIs rather than ports are used to demultiplex packets at the RSIP server, the RSIP server may still allocate mutually exclusive port numbers to the RSIP clients. If this does not happen, there is the possibility that two RSIP clients using the same IP address attempt an IPsec session with the same server using the same source port numbers.



For example, consider hosts X1 and X2 depicted above. Assume that they both are using public address Nb, and both are contacting an external server Y at port 80. If they are using IPsec but are not allocated mutually exclusive port numbers, they may both choose the same ephemeral port number to use when contacting Y at port 80. Assume client X1 does so first, and after engaging in an IKE negotiation begins communicating with the public server using IPsec.

When Client X2 starts its IKE session, it sends its identification to the public server. The latter's SPD requires that different identities use different flows (port numbers). Because of this, the IKE negotiation will fail. Client X2 will be forced to try another ephemeral port until it succeeds in obtaining one which is currently not in use by any other security association between the public server and any of the RSIP clients in the private network.

Each such iteration is costly in terms of round-trip times and CPU usage. Hence --and as a convenience to its RSIP clients--, an RSIP server may also assign mutually exclusive port numbers to its IPsec RSIP clients.

Despite proper allocation of port numbers, an RSIP server cannot prevent their misuse because it cannot examine the port fields in packets that have been encrypted by the RSIP clients. Presumably, if the RSIP clients have gone through the trouble of negotiating ports numbers, it is in their best interest to adhere to these assignments.

Appendix B: RSIP Error Numbers for IKE and IPsec Support

This section provides descriptions for the error values in the RSIP error parameter beyond those defined in [RSIP-P].

401: IPSEC_UNALLOWED. The server will not allow the client to use end-to-end IPsec.

402: IPSEC_SPI_UNAVAILABLE. The server does not have an SPI available for client use.

403: IPSEC_SPI_INUSE. The client has requested an SPI that another client is currently using.

Appendix C: Message Type Values for IPsec Support

This section defines the values assigned to RSIP message types beyond those defined in [RSIP-P].

22 ASSIGN_REQUEST_RSIPSEC

23 ASSIGN_RESPONSE_RSIPSEC

Appendix D: A Note on Flow Policy Enforcement

An RSIP server may not be able to enforce local or remote micro-flow policy when a client uses ESP for end-to-end encryption, since all TCP/UDP port numbers will be encrypted. However, if AH without ESP is used, micro-flow policy is enforceable. Macro-flow policy will always be enforceable.

Appendix E: Remote Host Rekeying

Occasionally, a remote host with which an RSIP client has established an IPsec security association (SA) will rekey [Jenkins]. SA rekeying is only an issue for RSIP when IKE port 500 is used by the client and the rekey is of ISAKMP phase 1 (the ISAKMP SA). The problem is that the remote host will transmit IKE packets to port 500 with a new initiator cookie. The RSIP server will not have a mapping for the cookie, and SHOULD drop the the packets. This will cause the ISAKMP

SA between the RSIP client and remote host to be deleted, and may lead to undefined behavior given that current implementations handle rekeying in a number of different ways.

If the RSIP client uses an ephemeral source port, rekeying will not be an issue for RSIP. If this cannot be done, there are a number of RSIP client behaviors that may reduce the number of occurrences of this problem, but are not guaranteed to eliminate it.

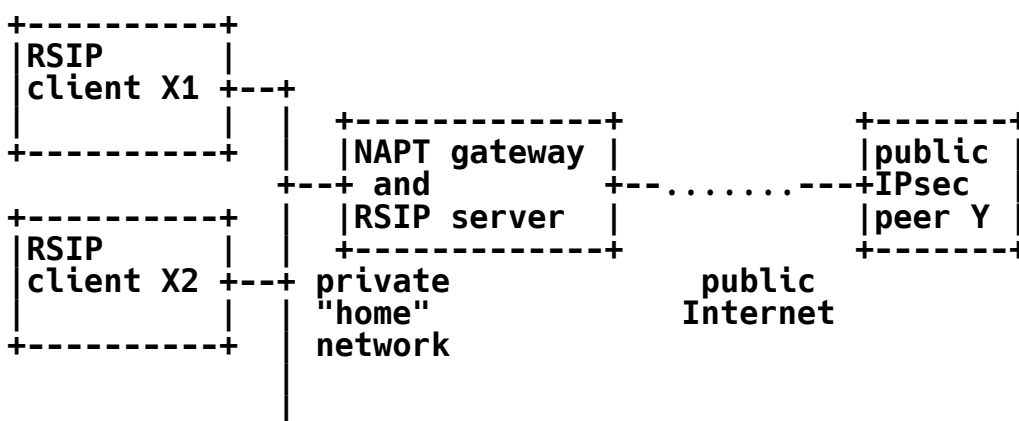
- The RSIP client's IKE implementation is given a smaller ISAKMP SA lifetime than is typically implemented. This would likely cause the RSIP client to rekey the ISAKMP SA before the remote host. Since the RSIP client chooses the Initiator Cookie, there will be no problem routing incoming traffic at the RSIP server.
- The RSIP client terminates the ISAKMP SA as soon as the first IPsec SA is established. This may alleviate the situation to some degree if the SA is coarse-grained. On the other hand, this exacerbates the problem if the SA is fine-grained (such that it cannot be reused by other application-level connections), and the remote host needs to initialize sockets back to the RSIP client.

Note that the unreliability of UDP essentially makes the ephemeral source approach the only robust solution.

Appendix F: Example Application Scenarios

This section briefly describes some examples of how RSIP may be used to enable applications of IPsec that are otherwise not possible.

The SOHO (small office, home office) scenario



Appendix G: Thoughts on Supporting Incoming Connections

Incoming IKE connections are much easier to support if the peer Y can initiate IKE exchanges to a port other than 500. In this case, the RSIP client would allocate that port at the RSIP server via `ASSIGN_REQUEST_RSAP-IP`. Alternatively, if the RSIP client is able to allocate an IP address at the RSIP server via `ASSIGN_REQUEST_RSA-IP`, Y could simply initiate the IKE exchange to port 500 at that address.

If there is only one address Nb that must be shared by the RSIP server and all its clients, and if Y can only send to port 500, the problem is much more difficult. At any given time, the combination of address Nb and UDP port 500 may be registered and used by only one RSIP system (including clients and server).

Solving this issue would require demultiplexing the incoming IKE connection request based on something other than the port and address combination. It may be possible to do so by first registering an identity with a new RSIP command of `LISTEN_RSIP_IKE`. Note that the identity could not be that of the IKE responder (the RSIP client), but that of the initiator (Y). The reason is that IKE Phase 1 only allows the sender to include its own identity, not that of the intended recipient (both, by the way, are allowed in Phase 2). Furthermore, the identity must be in the clear in the first incoming packet for the RSIP server to be able to use it as a demultiplexor. This rules out all variants of Main Mode and Aggressive Mode with Public Key Encryption (and Revised Mode of Public Key Encryption), since these encrypt the ID payload.

The only Phase 1 variants which enable incoming IKE sessions are Aggressive Mode with signatures or with pre-shared keys. Because this scheme involves the RSIP server demultiplexing based on the identity of the IKE initiator, it is conceivable that only one RSIP client at a time may register interest in fielding requests from any given peer Y. Furthermore, this precludes more than one RSIP client's being available to any unspecified peer Y.

Once the IKE session is in place, IPsec is set up as discussed in this document, namely, by the RSIP client and the RSIP server agreeing on an incoming SPI value, which is then communicated to the peer Y as part of Quick Mode.

The alternate address and port combination must be discovered by the remote peer using methods such as manual configuration, or the use of KX (RFC2230) or SRV (RFC2052) records. It may even be possible for the DNS query to trigger the above mechanisms to prepare for the incoming and impending IKE session initiation. Such a mechanism would allow more than one RSIP client to be available at any given

time, and would also enable each of them to respond to IKE initiations from unspecified peers. Such a DNS query, however, is not guaranteed to occur. For example, the result of the query could be cached and reused after the RSIP server is no longer listening for a given IKE peer's identity.

Because of the limitations implied by having to rely on the identity of the IKE initiator, the only practical way of supporting incoming connections is for the peer Y to initiate the IKE session at a port other than 500.

Full Copyright Statement

Copyright (C) The Internet Society (2001). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

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