Network Working Group M. Richardson
Internet-Draft SSW
Expires: August 10, 2002 D. Redelmeier
Mimosa
H. Spencer
SP Systems
February 9, 2002

A method for doing opportunistic encryption with IKE
draft-richardson-ipsec-opportunistic-06.txt

Status of this Memo

Copyright Notice

Table of Contents

Abstract

This document describes opportunistic encryption using IKE and IPsec. The objective is to allow encryption without any pre-arrangement specific to the pair of gateways involved. Each gateway administrator makes local arrangements to support opportunistic encryption. Once that is done, any two such gateways can communicate securely.

There are two large payoffs. First, if many machines must communicate securely, this approach reduces administrative overhead to order N (number of gateways), rather than the N squared cost to configure each tunnel separately. Second, it becomes possible to make secure communication the default, even when the partner is not known in advance.

There are naturally some risks and some costs, which are described.

This document is offered up as an Informational RFC.

1. Introduction

1.1 Motivation

The objective of opportunistic encryption is to allow encryption without any pre-arrangement specific to the pair of gateways involved. Each gateway administrator makes local arrangements -- specifically, adds public key information to DNS records -- to support opportunistic encryption. Once that is done, any two such gateways can communicate securely.

This document describes opportunistic encryption as designed and (to date, partially) implemented by the Linux FreeS/WAN project. For project information, see www.freeswan.org.

The Internet Architecture Board (IAB) and Internet Engineering Steering Group have taken a strong stand that the Internet should use powerful encryption to provide security and privacy [4]. This project attempts to put this policy into practice by providing practical means to implement them.

The project believes that the IPsec protocols are the best chance to do that, because they are standardized, widely available and can often be deployed very easily, without changing hardware or software or retraining of users. However, the extension to opportunistic encryption seems necessary, both to help control administrative overheads and to get IPsec more widely used.

The use of "opportunistic encryption" offers the "fax effect". As each person installs one for their own use, it becomes more valuable for their neighbors to install one too, because there's one more person to use it with. The software automatically notices each newly installed box, and doesn't require a network administrator to reconfigure it.

This document describes the infrastructure needed to support this effort.

The term S/WAN is a trademark of RSA Data Systems, and is used with permission by this project.

1.2 Types of network traffic

To aid in understand the relationship between security processing and IPsec we divide network traffic into four categories:

deny networks to which traffic is always forbidden

permit networks to which traffic in the clear is desired

opportunistic tunnel networks to which encryption should be done if possible, but sent in the clear otherwise

configured tunnel networks to which encryption must be done, and never sent in the clear

This document describes the third category. The first two categories are provided by traditional firewall devices. The fourth category is presently the offering of many Virtual Private Network (VPN) devices.

Category one, denied traffic by IP address, requires no authentication.

Category two, permitted traffic by IP address, requires no authentication. This is the normal default on the Internet.

Category four, encrypt traffic or drop it, requires authentication of the end points. As the number of end points is typically bound and is typically under the authority of a single authority, arranging for exchange of authentication material, while difficult, does not require any new technology. The mechanism described here provide an additional way to provide the authentication materials, notably a public key method that does not require deployment of an X.509 based infrastructure.

1.3 Authentication of Opportunistic Encryption

Opportunistic encryption involves creating tunnels with other nodes with are essentially strangers. This is done without any prior bilateral arrangement. There is therefore the difficult question of how does one know who one is talking to.

One answer is that one does not know who one is talking to. No useful authentication can be done, so do not even try. This mode of operation has been given the name "anonymous encryption".

Although a useful mode, it is not the goal of this project. It is a useful starting point, but the system should permit additional layers of trust to be built upon this system. In the described system, the anonymous encryption case is what results without DNSSEC. Were anonymous encryption the end goal, simpler methods are available to achieve this goal.

However, an essential premise of building private connections with strangers is that datagrams received through these opportunist tunnels datagrams are no more special than datagrams that arrived in

the clear.

Unlike in a VPN scenario, these datagrams should not be given any special exceptions when it comes to auditing, further authentication or firewalling.

On the outbound side, when initiating opportunistic encryption, it becomes a local matter what to do if one fails to setup a tunnel. It may be that the packet goes out in the clear, or it may be dropped. This is a local configuration matter.

In sum, we gain wider privacy (for the Internet at large) at minimal cost: the cost is the need to reassess assumptions about the relationship between IPsec authentication and further local access control.

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in [5]

2. Overview

2.1 Reference diagram

The following network diagram is used in the rest of this document as the canonical diagram

```
                      [Q]  [R]          AS2
                       .    .
   [A]----+----[SG-A]...+....+....[SG-B]-------[B]
      AS1 |            . PI .
   [D]----+----[SG-D]...+    +....[C]    AS3
```

Figure 1: Reference Network Diagram

In this diagram, there are four end-nodes: A, B, C and D. There are three gateways, SG-A, SG-B, SG-D. A, D, SG-A and SG-D are part of the same administrative authority. SG-A and SG-D are on two different exit paths from organization 1. SG-B/B is an independent organizations. Nodes Q and R are nodes that are on the Internet. PI is the Public Internet ("The Wild").

2.2 Terminology

The following terminology is used in this document:

security gateway: a system that performs IPsec tunnel mode encapsulation/decapsulation. [Gx] in the diagram

Alice: node [A] in the diagram. When an IP address is needed, this is 192.1.0.65.

Bob: node [B] in the diagram. When an IP address is needed, this is 192.2.0.66.

Carol: node [C] in the diagram. When an IP address is needed, this is 192.1.1.67.

Dave: node [D] in the diagram. When an IP address is needed, this is 192.3.0.68

SG-A Alice's security gateway. Internally it is 192.1.0.1, externally it is 192.1.1.4.

SG-B Bob's security gateway. Internally it is 192.2.0.1, externally it is 192.1.1.5.

SG-D Dave's security gateway. Also Alice's backup security gateway. Internally it is 192.3.0.1, externally it is 192.1.1.6.

- a single dash represents clear-text datagrams

= an equals sign represents phase 2 (IPsec) cipher-text datagrams

a hash sign represents phase 1 (IKE) cipher-text datagrams

. a period represents an untrusted network of unknown type

configured tunnel: Contrast with opportunistic tunnel. A tunnel that is directly/deliberately/hand configured on participating gateways. Configured tunnel are typically given a higher level of trust than opportunistic tunnels.

road warrior tunnel: a configured tunnel connecting one node with a fixed IP address and one node with a variable IP address. A road warrior (RW) connection must be initiated by the variable node, since the fixed node does not know what the address for the "road warrior" currently is.

phase 1 SA an ISAKMP/IKE security association, sometimes also referred to as a keying channel.

phase 2 SA an IPsec security association

tunnel another term for a set of phase 2 SA

NAT Network Address Translation (see [18])

NAPT Network Address and Port Translation (see [18])

anonymous encryption: The process of encrypting a session without any knowledge of who the receiving party is. That is, no authentication of identities is done.

opportunistic encryption: The process of encrypting a session with knowledge of who the receiving party is.

lifetime: The negotiated period in seconds (bytes or datagrams) which a security association will remain alive before needing

to be re-keyed.

lifespan: The effective time which a security association remains useful. A security association with a lifespan shorter than its lifetime would be removed when no longer needed. A security association with a lifespan longer than its lifetime would need to be re-keyed one or more times.

2.3 Model of Operation

The Opportunistic Encryption security gateway ("OE gateway") is a superset of a regular gateway node as described in [2] section 2.4 and [3]. It will also be capable of performing IPsec security gateway functions as described in [6]. The algorithm described here provides a way to determine on a datagram by datagram basis whether or not to apply an encrypting tunnel algorithm to the datagram. Thus, two important things that must be determined are whether or not to tunnel and to which destination node the tunnel should be made.

2.3.1 Tunnel Authorization

The decision as to whether or not to create a tunnel is determined on a per-destination address basis. Upon receiving a packet with a destination address never before seen, the OE gateway performs a lookup in DNS for an authorization record. (see Section 5.2) This record is located using the IP address to perform a search in the in-addr.arpa (IPv4) or ip6.arpa (IPv6) maps. The presence of an authorization record indicates the desire for a tunnel to be formed.

2.3.2 Tunnel End-point discovery

The record further provides the address or name of the end-point which should be used.

The record may also provide the public RSA key of the tunnel end point itself. This is provided for efficiency only - if this is not present, then the address or name provided is used to perform a second lookup to find a KEY Resource Record.

DNSSEC ([14]) SHOULD be used to provide origin and integrity protection of the Resource Records involved.

2.3.3 Caching of authorization results

The OE gateway MUST maintain a list of source/destination pairs for which Opportunistic Encryption has been attempted for a period of time. Successful discovery of a tunnel end point SHOULD result in

creation of a new security association bundle between the OE gateway and the discovered tunnel end point.

Failure to discover an appropriate authorization MUST result in creation of a forwarding policy entry to either drop or transmit in the clear future datagrams. This negative cache is necessary so as to avoid repeatedly looking up the same information, a possibly lengthly process.

OE gateways MUST time out all cache entries (both positive and negative ones) periodically. This is done both to remove entries which are no longer necessary, and to permit changes in authorization policy to be discovered.

3. Specification

The OE gateway is modeled to have a forwarding plane and a control

plane. They are connected by a control channel such as PF_KEY. The forwarding plane performs per-datagram operations. The control plane contains a keying daemon such as ISAKMP/IKE and performs all authorization, authentication and key derivation functions.

3.1 Datagram State Machine

Let the OE gateway maintain a collection of objects -- a superset of the Security Policy Database specific in [6]. For each combination of source and destination address, one may find an SPD object in one of six states. Each datagram which is a candidate for forwarding has is used to index into this database to determine how it should be treated.

3.1.1 Nonexistent policy

No entry may exist. An entry is created with an initial state of "Hold policy". A request for keying material is sent to the keying daemon. The datagram is not forwarded, rather it is attached to the SPD entry as the "first" datagram and retained for eventual transmission in a new state.

3.1.2 Hold policy

A request for keying has been previously made. If the interface to the keying system is lossy (PF_KEY, for instance, can be) a mechanism to retransmit the keying request, rate limited to less than 1 request per second SHOULD be made. The datagram is not forward. The datagram is attached to the SPD entry as the "last" datagram and retained for eventual transmission. If there was previously a datagram so stored, then it is discarded.

3.1.3 Pass-through policy

The datagram is forwarded using the normal forwarding table. This state is entered only via command from the keying daemon. Upon entering this state, the "first" and "last" datagrams are forwarded.

3.1.4 Drop policy

The datagram is discarded. This state is entered only via command from the keying daemon. Upon entering this state, the "first" and "last" datagrams are also discarded.

3.1.5 Encrypt policy

The datagram is encrypted using the indicated Security Association Database (SAD) entry. This state is entered only via command from the keying daemon. Upon entering this state, the "first" and "last" datagrams are sent.

If the associated SAD entry expires (due to byte, packet or time limits) then the entry returns to the Hold policy, causing an expire message to communicated to the keying daemon. All states may be created directly by the keying daemon while acting as a responder.

3.2 Keying State Machine - Initiator

Let the keying daemon maintain a collection of objects -- let them be called "connections" (or "conn"s for short). There are two categories of connection objects: classes and instances. A class connection represents an abstract policy - what could be. An instance represents an actual connection, what is in fact implemented at the time.

Let there be two further subtypes of connections: keying channels (aka Phase 1 SAs) and data channels (aka Phase 2 SAs). Each data channel object may have a corresponding SPD and SAD entry maintained by the Datagram State Machine.

For the purposes of Opportunistic Encryption, there MUST at least be connection classes known as "deny", "always-clear-text", "OE-permissive", "OE-paranoid". The later two connection classes defines a set of source and/or destination addresses for which Opportunistic Encryption will be attempted. There are a number of additional places where the administrator MAY set policy options. An implementation MAY create additional connection classes so that these policies may be enacted in a more fine grained fashion.

{should the additional classes be given names here? - ed.}

The simplest system may need only the "OE-permissive" connection, and would list its own (single) IP address as the source address of this policy, and the wild-card address 0.0.0.0/0 as the destination address. That is, the simplest policy is to try Opportunistic Encryption will all destinations.

The distinction between permissive and paranoid OE use will become clear in the state transition differences. In general, a permissive OE will fail to clear text packets, while a paranoid OE will fail to not sending any data.

In this description of the keying machines state transitions, the states associated with the keying system itself are omitted. This is done for two reasons: they are best documented in the keying system ([7], [8] and [9] for ISAKMP/IKE), and the details are keying system specific. Opportunistic Encryption is not dependent upon any

specific keying protocol, but this document does provide set some requirements for those using ISAKMP/IKE to assure inter-operability.

The state transitions that may be involved in communicating with the forwarding plane are omitted. PF_KEY and similar protocols have their own state of states required for message sends and completion notifications.

Finally, the retransmits and recursive lookups that are normal for DNS are not included in this state machine.

3.2.1 Nonexistent connection

There is no connection for a given source/destination address pair. Upon receipt of a request for keying material for this particular source/destination pair, a search is made through the connection classes to determine the most appropriate policy. Upon determining an appropriate connection class, then an instance object is created of that type. Both of the OE types results in a Potential OE connection.

Failure to find an appropriate connection class results in an administrator defined default.

In each case, when an appropriate class is found for the new flow then an instance connection is made of the type which matched.

3.2.2 clear-text connection

A transition is made from the empty connection to this state when an instance of the always-clear-text class is instantiated, or when an OE-permissive connection fails. During the transition, a pass-through policy object is created in the forwarding plane for the appropriate flow.

The only way to leave this state is due to a timeout, see expiry connection.

3.2.3 Deny connection

A transition is made from the empty connection to this state when an instance of the deny class is instantiated, or when an OE-paranoid connection fails. During the transition, a pass-through policy

object is created in the forwarding plane for the appropriate flow.

The only way to leave this state is due to a timeout, see expiry connection.

3.2.4 Potential OE connection

A transition is made from the empty connection to this state when an instance of either OE class is instantiated. During the transition to this state, a hold policy object is created in the forwarding plane for the appropriate flow.

In addition, when transitioning into this state, DNS lookup(s) are done in the reverse map for a TXT delegation resource record. (see Section 5.2) The destination address of the flow is used as the lookup key.

There are three ways to exit this state: due a timeout in the DNS lookup, and via positive or negative replies as to the existence of the TXT delegation resource record.

If there is a resource record found, and it is properly formatted, and if DNSSEC is enabled - the signature has been vouches for (either through local confirmation or via trusted path to a recursive DNSSEC server), then there is a transition to the Pending OE connection state. (Note that if the public key is not presented in the TXT delegation record, then it must be looked up as well as a sub-state. The DNS lookups are not considered a success until all have completed successfully)

If there is no resource record found, or DNS times out then it is to be considered that this is not an OE capable receiver. If this was an OE-paranoid instance, then there is a transition to the deny connection. If this was an OE-permissive instance, then there is a transition to the clear-text connection.

If the resource record is found but is misformed, or if DNSSEC has been enabled and reports a failure to authenticate, then there should be a transition to the deny connection. This fact should be logged. If the administrator wishes to override this, then an always-clear class can be installed for this flow. An implementation MAY make this situation a new class.

An implementation SHOULD also provide an additional administrative control on delegation records and DNSSEC. If this control is enabled, then delegation records which are not secured with DNSSEC may only point to themselves. When this option is enabled, an active attack on DNS will be unable to redirect packets to other than the

original destination.

3.2.5 Pending OE connection

A transition is made from the Potential OE connection to this state when it has been determined that is is worth attempting to do keying for this connection. Upon entering this state, an attempt to

initiate keying to the gateway provided is started.

One exits from this state either with a successfully created IPsec SA, or with a failure of some kind. Successful SA creation results in a transition to the Key connection state.

There are three failures which are distinguished. They are clearly not the only possible failures from keying, but these are the ones that have caused the most problems.

Note that if there were multiple gateways available in the TXT delegation records, then a failure can only be declared after all have been tried. Further, creation of a phase 1 SA does not constitute success.

The first failure is when an ICMP port unreachable is consistently received without any other communication, or when there is silence from the remote end. This likely means that the gateway is either not alive, or that the keying daemon is not functional. For an OE-permissive connection, transition to the clear-text connection, but with a rather low timeout. The gateway may be in the process of rebooting, etc. For an OE-pessimistic connection, transition to the deny connection, again with a low timeout.

If a phase 1 SA is created, but there is either no response to the phase 2 proposal, or a negative notify is received, then the remote gateway is not prepared to do OE at this time. As before transition to clear-text/deny based upon connection class, but this time with a relatively long timeout.

The third type of failure is when there is signature failure authenticating the remote gateway. In this case, again transition to clear-text/deny based upon the connection class, but make the timeout depend upon the remaining time to live in the DNS. (Note that DNSSEC resource records have a different expiry time than non-signed records) One possibility is that there has been a key roll-over, but that DNS has no caught up.

3.2.6 Keyed connection

A transition is made from the Pending OE connection to this state

when session keying material (aka the phase 2 SA) has been formed. An Encrypt Policy is created in the forwarding plane for this flow.

There are three ways to exit this state. The first is by receipt of an authenticated delete message (via the keying channel) from the peer. This is normal teardown, and results in a transition to Expired connection.

The second way is by expiry of the forwarding plane keying material. This causes a re-key operation to be started with a transition back to Pending OE connection. In general, the soft expiry will occur with sufficient time left for the keys to continue to be used. Note that a re-key can fail, which may result in the connection failing to clear-text or deny.

The third way is via receipt of an initiate from the remote gateway, via receipt of a indication from the forwarding plane of having received unknown SPI from the gateway, or an ICMP from the remote gateway indicating an unknown SPI. Each of these things should be considered suspicious, but provide a hint that the remote gateway has rebooted or restarted. A cautious (rate-limited) attempt to re-key the connection should be done.

3.2.7 Expiring connection

Each of the deny, clear-text, and keyed connections will periodically be placed into this sub-state. See Section 3.4 for more details. The forwarding plane is queried for last use time of the appropriate policy. If the use time is relatively recent, then the state returns to the previous deny, clear-text or keyed connection state. If not, then it enters the expired connection state.

Note that both outgoing SPD and incoming SAD must be queried as some flows may be very unidirectional.

3.2.8 Expired connection

Entry to this state is due to no datagrams being forwarded via the appropriate SPD and SAD objects. The objects in the forwarding plane are removed (logging any final byte and packet counts if appropriate) and the connection instance in the keying plane is deleted.

3.3 Keying State Machine - Responder

The responder the identical set of objects as the initiator.

The responder gets its first indication that something is happening when it receives an invitation to create a keying channel from an

initiator. The responder should check the incoming address against its connection class list. If this is an address to which it would never consider doing OE (i.e. an always clear or a deny) then it MAY ignore the invitation and remain silent. (There may be other reasons for it to reply, such as support for mobile nodes). {problem with this statement is that the responder actually has no idea what policy to pick, because it does not if the initiator is acting on behalf of some other node. -ed}

If the responder decides that this is a potential peer, then an instance object is created and a transition to Unauthenticated OE peer is made.

3.3.1 Unauthenticated OE peer

Upon entering this state, a DNS lookup is done for a KEY record for the initiator. This is done in the reverse map for a KEY record for the initiator if the initiator has offered an ID_IPV4_ADDR, and in the forward map if the initiator has offered an ID_FQDN type. (See [7] section 4.6.2.1.)

This state is exited upon successful receipt of a KEY from DNS, and use of it to verify the signature of the initiator. The public that is retrieved should be stored in stable storage for an administratively defined period of time (typically several months if possible). If a key has previously been stored on disk, then the returned key should be compared to what has been received, and the key considered valid only if they match.

Successful authentication of the peer results in a transition to Authenticated OE Peer.

Note that this state generally occurs in the middle of the key negotiation protocol.

3.3.2 Authenticated OE Peer

The peer will eventually propose one or more phase 2 SAs. The source and destination address in the proposal are used to initialize the still empty connection state using the connection class table. A search for an identical connection object MUST be made at this point.

If an identical connection is found, then delete the instance that has been created, and transition the existing object to the Pending OE connection state.

If an identical connection is not found, then transition according to the rules given for the initiator.

Note that if the initiator is in OE-paranoid mode and the responder is in either always-clear-text or deny, then no communication is possible according to policy. {this suggests that there is a further set of states, which are "accept OE, but do not initiate it instead always be clear-text or deny.}

3.4 Renewal and Teardown

3.4.1 Aging

A potentially unlimited number of tunnels may exist. In practice, a few tunnels are used intensively for short periods of time. Unused tunnels MUST therefore be torn down. Detecting when they are no longer in use is the subject of this section.

There are two methods in which the tunnel may be removed: by expiring or with explicit deletion.

Explicit deletion is done with an IKE Delete message. To do this requires that both ends maintain the key channel (phase 1 ISAKMP SA), as the deletes MUST be authenticated. An implementation which refuses to maintain the keying channel SA will be unable to use this method.

In the expiry method, the tunnel is simply allowed by the IKE daemon to expire normally, without attempting to re-key it.

Regardless of which method is used, a method is required to determine if the tunnel is still in use. This is a local matter, but the following criteria are what is used by the FreeSWAN project. This criteria is currently implemented in the key management daemon, but could also be implemented at the SPD layer using an idle timer.

+ a short initial (soft) lifespan of 1 minute is set. This is done since many net flows in fact last only a few seconds.

+ at the end of the lifespan, a check is made to see if the tunnel was used by traffic in either direction during the last half of this period. If so, assign a longer tentative lifespan, of 20 minutes, after which, look again. If the tunnel is not in use then close the tunnel.

These timeouts are implemented in by the Expiring state in the key management system (see Section 3.2.7). The timer given above may in fact be present in the forwarding plane, but it must, in this case be resettable. In the absence of any traffic counts in the forwarding plane, then the forwarding plane policy can be changed temporary to a hold policy. In this case, receipt of an acquire message would cause

the original encrypt policy to be re-instated again.

The tentative lifespan is independent of re-keying; it is just the time when the tunnel's future is next considered. (The term lifespan is used here rather than lifetime for this reason.) This should happen reasonably frequently, unlike re-keying, which is costly and shouldn't be too frequent.

A multi-step back-off algorithm is not considered worth the effort.

If the security gateway and the client host are one and the same,

tunnel teardown decisions MAY pay attention to TCP connection status, as reported by the local TCP layer. A still-open TCP connection is almost a guarantee that more traffic is coming, while the demise of the only TCP connection through a tunnel is a strong hint that no more traffic will transit.

3.4.2 Teardown and Cleanup

Teardown should always be coordinated with the other end. This means interpreting and sending Delete notifications. There is some detailed state in the Expired Connection state of the key manager that relates to retransmits of the Delete notifications, but this is considered to be a keying system detail.

On receiving a Delete for the outbound SAs of a tunnel (or some subset of them), tear down the inbound ones too, and notify the other end with a Delete. If a Delete is received for a tunnel which is no longer in existence, then two Delete messages have crossed paths. Ignore the Delete - the operation has already been done. Do not generate any messages in this situation.

Tunnels need to be considered as bidirectional entities, even though the low-level protocols don't think of them that way.

When the deletion is initiated locally, rather than as a response to a received Delete, send a Delete for (all) the inbound SAs of a tunnel. If no responding Delete is received for the outbound SAs, try re-sending the original Delete. Three tries spaced 10s apart seems a reasonable level of effort. A failure for the other end to respond at this point likely indicates that no further communication will be possible in any case.

After re-keying, transmission should switch to using the new SAs (ISAKMP or IPsec) immediately, and the old leftover SAs should be cleared out promptly (and Deletes sent) rather than waiting for them to expire. This reduces clutter and minimizes confusion.

Since there is only one keying channel per remote IP address, the question of whether a Delete notification has appeared on a ''suitable'' phase 1 SA does not arise.

4. Impacts on IKE

4.1 ISAKMP/IKE protocol

The IKE wire protocol needs no modifications. The major changes are implementation issues relating to how the proposals are interpreted, and from whom they may come.

As Opportunistic Encryption is designed to be useful between peers without prior operator configuration, an IKE daemon must be prepared to negotiate phase 1 SAs with any node. This may require a large amount of resources to maintain cookie state, as well as large amounts of entropy to for nonces, cookies, etc.

The major changes to support Opportunistic Encryption are at the IKE daemon level. These changes relate to handling of key acquisition requests, lookup of public keys and TXT records, and interactions with firewalls and other security facilities that may be coresident on the same gateway.

4.2 Gateway discovery process

In a typical configured tunnel situation, the address of SG-B is provided via configuration. Furthermore, the mapping of SPD entry to gateway is typically a 1:1 mapping. When the 0.0.0.0/0 SPD entry technique is used, then the mapping to a gateway is determined by the reverse DNS records.

The need to do a DNS lookup and wait for a reply will typically introduce a new state and a new event source (DNS replies) to IKE. Although a synchronous DNS request can be done for proof of concept, this will not scale very well.

Use of an asynchronous DNS lookup will also permit overlap of DNS lookups with protocol steps.

4.3 Self identification

SG-A will have to establish its identity. This is done by use of an IPv4 ID in phase 1.

There are many situations where the administrator of SG-A may not be able to control their reverse DNS. Typical situations include dialup connections and most residential-type broadband Internet access (ADSL, cable-modem). In these situations, a fully qualified domain name which is under the control of SG-A's administrator may be used. The FQDN ID should be used in phase 1. See Section 5.3 for more details and restrictions.

4.4 Public key Retrieval process

Upon receipt of a phase 1 SA proposal with either an IPv4 (IPv6) ID, or an FQDN ID, an IKE daemon will need to examine local caches and configuration files to determine if this is part of a configured tunnel. If none is found, then the implementation should attempt to retrieve a KEY record from the reverse DNS (in the case of an IPv4/IPv6 ID), or from the forward DNS in the case of FQDN ID.

It is reasonable that if other non-local sources of policy are used (COPS, LDAP, ...) that they be consulted concurrently, but that some clear ordering of policy be provided. Note that due to variances in latency, one must wait for positive or negative replies from all sources of policy before making any decisions.

4.5 Interactions with DNSSEC

The present implementation does not use DNSSEC to explicitly verify the authenticity of zone information, or use the NXT records to provide authentication of the absence of a TXT or KEY record. These are important considerations for practical use.

In practice the verification of the DNSSEC SIG and NXT records will typically be done by a trusted DNS server. This process may be co-located, or reachable via a trusted path.

4.6 Recommended proposal types

4.6.1 Phase 1 IDs

Main mode MUST be used.

The initiator MUST offer at least one proposal using some combination of: 3DES, HMAC-MD5 or HMAC-SHA1, DH group 2 or 5. Group 5 SHOULD be proposed first. [11]

The initiator MAY offer additional proposals, but the cipher MUST not be weaker than 3DES. The initiator SHOULD limit the number of proposals such that the IKE datagrams do not need to be fragmented.

The responder MUST accept one of the proposals. If any configuration of the responder is required then the responder is not acting in an opportunistic way.

SG-A SHOULD use an ID_IPV4_ADDR, of the external interface of SG-A for phase 1. (There is an exception, see Section 5.3) The authentication method SHOULD be RSA public key signatures.

The RSA key for SG-A SHOULD be placed into a DNS KEY record in the reverse space of SG-A. (i.e. using in-addr.arpa.)

4.6.2 Phase 2 IDs

SG-A SHOULD propose a tunnel between Alice and Bob, using 3DES-CBC mode, MD5 or SHA1 authentication. Perfect Forward Secrecy SHOULD be specified.

Authorization for SG-A to act on Alice's behalf is determined by looking for a TXT record in the reverse map at Alice's address.

5. DNS issues

5.1 Use of KEY record

In order to establish its own identity, SG-A and SG-B must publish their public keys in their reverse DNS. This is just done via DNSSEC's KEY record. See section 3 of RFC 2535 [14].

For example:

KEY 0x4200 4 1 AQNJjkKlIk9...nYyUkKK8

0x4200 The flag bits, indicating that this key is prohibited for use confidentiality (it authenticates the peer only, DH is used for confidentiality), and that this key is associated with the non-zone entity whose name is the RR owner name. No other flags are set.

4 This indicates that this key is for use by IPsec

1 An RSA key is present

AQNJjkKlIk9...nYyUkKK8 the public key of the host as described in [15]

5.2 Use of TXT delegation record

A TXT record is published by Alice (Bob) to provide authorization for SG-A (SG-B) to act on their behalf. This record is located in the reverse DNS (in-addr.arpa) for them. The reverse DNS SHOULD be secured by DNSSEC, when it is deployed. DNSSEC is required to defend against active attacks.

The mechanism described here provides a new way to change the routing of IP datagrams other than the routing system. As a matter of local policy, an implementation MAY ignore these records unless it is certain that they have been signed with DNSSEC.

If Alice's address is P.Q.R.S, then she can authorize another node to act on her behalf by publishing records at:

S.R.Q.P.in-addr.arpa

The contents of the resource record are expected to be a string that follows the following syntax, as suggested in [13]. (Note that the reply to query may include other resource records from other applications may also be returned.)

```
X-IPsec-Server(P)=A.B.C.D KEY
```

Figure 2: Format of reverse delegation record

P: the P specifies a precedence for this record. This is similar to MX record preferences. Lower numbers have stronger preference.

A.B.C.D: specifies the IP address of the Security Gateway for this client machine.

KEY: is the encoded RSA Public key of the Security Gateway. This

is provided here to avoid a second DNS lookup. If this field is absent, then a KEY resource record should be looked up in the reverse map of A.B.C.D.

In the case where Alice is located at a public address behind a security gateway that has no fixed address (or no control over its reverse map), then Alice may delegate to a public key by domain name:

```
X-IPsec-Server(P)=@FQDN KEY
```

Figure 3: Format of reverse delegation record (FQDN version

P: is as above.

FQDN specifies the FQDN that the Security Gateway will identify itself with.

KEY: is the encoded RSA Public key of the Security Gateway.

If there is more than one such TXT record with strongest (lowest numbered) precedence, one Security Gateway is picked arbitrarily from those specified in the strongest-preference records. All keys for that Security Gateway are made available as candidates for signature checking. This mechanism is required to permit roll-over of signature keys in a seamless fashion.

5.2.1 Choice of TXT record

It has been suggested to use the OPT, CERT, KEY or KX records instead of a TXT record.

They KEY RR has a Protocol field which could be used to indicate use for a new protocol, and an Algorithm field which could be used to indicate different contents in the key data. However, the KEY record is not clearly intended for storing what are really authorizations, it is just for identities. Other uses have been discouraged.

OPT resource records, as defined in [12] are not intended to be used for storage of information. They are not to be loaded, cached or forwarded. They are therefore inappropriate for use here.

CERT records [16] can encode almost any set of information. A custom Type code would be used permitting any suitable encoding to be stored, not just X.509. The certificate according to the RFC, are

signed internally, which may add undesirable and unnecessary bulk. Larger DNS records may require TCP transfers instead of UDP ones.

At the time of protocol design, the CERT RR was not widely deployed and could not be counted upon. Use of CERT records will be investigated, and may result in a future revision of this document.

KX records are ideally suited for this use, but have not been deployed in any known DNS implementations.

5.3 Use of FQDN IDs

Unfortunately, not every administrator has control over the contents of the reverse-map. The only case where we can work around this is where the initiator (SG-A) has no suitable reverse map. In this case, the authorization record present in the reverse map of Alice may refer to a FQDN instead of an IP address.

In this case, the client's TXT record gives the fully qualified domain name (FQDN) in place of its security gateway's IP address. The initiator should use the ID_FQDN ID-payload in phase 1. A forward lookup for a KEY record on the FQDN must yield the initiator's public key.

This method can also be used when the external address of SG-A is dynamic.

If SG-A is acting on behalf of Alice, then Alice must still delegate authority for SG-A to do so in her reverse map. When Alice is SG-A (i.e. Alice is acting as an end-node) then there is no need for

this. See Figure 3

6. Network Address Translation interaction

There are no fundamentally new issues for getting opportunistic encryption to work in the presence of network address translation. Rather there are just the regular IPsec issues with NAT traversal.

There are several situations to consider for NAT:

6.1 Colocated NAT/NAPT

If SG-A is also performing Network Address Translation on behalf of Alice, then the packet should be translated prior to being subjected to opportunistic encryption. This is in contrast to typical configured tunnel which often exist to bridge islands of private network address space. SG-A will use the translated source address for phase 2, and so SG-B will look that address up to confirm SG-A's authorization.

In the case of NAT (1:1), the address space into which the translation is done MUST be globally unique, and so control over the reverse map is assumed to be a given. Placing of TXT records is possible.

In the case of NAPT (m:1), the address will be SG-A. The ability to get KEY and TXT records in place will again depend upon whether or not there is administrative control over the reverse map. This is identical situations involving a single host acting on behalf of itself. FQDN style can be used to get around a lack of a reverse map for initiators only.

6.2 SG-A behind NAT/NAPT

If there is a NAT or NAPT between SG-A and SG-B, then normal IPsec NAT traversal rules would apply. In addition to the transport problem which can be solved via proposals like [10], there is the issue of what phase 1 and phase 2 IDs to use. While FQDN could be used during phase 1 for SG-A. An appropriate ID for phase 2 permits SG-B to determine that SG-A was in fact authorized to speak for Alice.

This is only possible if Alice actually has a public IP. It is a somewhat unusual case where a public network exists behind SG-A, while SG-A itself is behind a NAPT. This may occur with mobile networks of various kinds that occasionally wind up behind a NAPT.

6.3 Bob is behind a NAT/NAPT

If Bob is behind a NAT (perhaps SG-B), then there is in fact no way

for Alice to address a packet to Bob. Not only is opportunistic encryption impossible, but it is also impossible for Alice to initiate any communication to Bob. It may be possible for Bob to initiate in such a situation.

7. Host implementations

When Alice and SG-A are components of the same system, then this is considered to be a host implementation. The scenario remains unchanged with respect to packet sequence.

Components marked Alice are simply the upper layers (TCP, UDP, the application), and SG-A is the IP layer.

Note that tunnel mode is still recommended.

As Alice/SG-A are acting on behalf of themselves, no TXT based delegation record is necessary for Alice. She can rely on a FQDN in a forward map.

8. Multihoming

If there are multiple paths between Alice and Bob (such as illustrated in the diagram with SG-D) then additional DNS records are required to establish authorization.

In the diagram in Figure 1, Alice has two ways to exit her network: SG-A and SG-D. Previously SG-D has been ignored. Postulate that there are routers between Alice and her set of security gateways (denoted by the + signs and the marking of an autonomous system number for Alice's network). Datagrams may therefore travel to either SG-A or SG-D en route to Bob.

As long as all network connections are in good order it does not matter how datagrams exit Alice's network. When they reach either security gateway, the security gateway will find the TXT delegation record in Bob's reverse map, and establish an SA with SG-B.

SG-B has no problem establishing that either of SG-A or SG-D may speak for Alice, as Alice has published two equally weighted TXT delegation records:

```
X-IPsec-Server(10)=192.1.1.5 AQMM...3s1Q==
X-IPsec-Server(10)=192.1.1.6 AAJN...j8r9==
```

Figure 4: Multiple gateway delegation example

Alice's routers can now happily do any kind of load sharing that they might wish to do. SG-A and SG-D will both send datagrams addressed to Bob through their tunnel to SG-B.

If Alice wishes to prefer one gateway over another, then Bob should, upon finding that it has connections to two gateways en route to Alice, prefer the one with the lower precedence.

If the precedences are the same, then SG-B has a more difficult time. It must decide which of the two tunnels to use. SG-B has no information about which link is less loaded, nor which security gateway has more cryptographic resources available. SG-B in fact has no knowledge of whether both gateways are even reachable.

The Public Internet's default free zone may well know a good route to Alice, but the datagrams that SG-B creates must be addressed to

either SG-A or SG-D; they can not be addressed to Alice directly.

There are a number of choices which SG-B may do:

1. It can ignore the problem and round robin among the tunnels it has. This will cause losses during times when one or the other security gateway is unreachable. If this worries Alice, she can change the weights in her TXT delegation records.

2. It can always send to the gateway that it most recently received from. This assumes that routing and reachability is symmetrical.

3. It can listen to BGP information from the Internet to decide which system is currently up. This is clearly a much more complicated thing to do, but if SG-B is already doing this because it is participating in the BGP peering system to announce Bob, the results data may already be available to it.

4. It can refuse to negotiate the second tunnel.

5. It can silently replace the first tunnel with the second one, while still being willing to accept datagrams from either SG-A or SG-D.

These are decisions left to local policy. Note that even if SG-B has perfect knowledge about reachability of SG-A and SG-D, Alice may not be reachable from one or other of these security gateways due to internal reachability issues.

FreeS/WAN implements option 5, but plans to implement option 2 in the future.

9. Failure modes

9.1 DNS failures

If a DNS server fails to respond, then it is a local policy decision whether or not to permit communication in the clear. It should be clear that mounting a denial of service attack on the DNS server responsible for a particular network's reverse map is an easy thing to do. Such an attack may cause all communication with that network to go in the clear. Please note that this is an active attack.

At the same time, there are still a very large number of networks that do not have properly configured reverse maps. Further, the effect of the above denial of service attack, if the policy is not to communicate, is that the target network becomes isolated. This is why this decision MUST be a matter of local policy.

9.2 DNS configured, IKE failures

In this situation, DNS records claim that opportunistic encryption should occur, but the target gateway either does not respond on port 500, or refuses the proposal. This may be due to a crash/reboot, due to misconfiguration, a firewall filtering port 500.

The receipt of ICMP port, host or network unreachable messages should

be taken as a sign that there is a potential problem, but MUST NOT cause communication to fail immediately.

It is recommended that in these situations that a clear log be produced about the problem. A local policy should dictate if communication is then permitted in the clear at this point.

9.3 System reboots

Tunnels sometimes go down because the other end crashes, or disconnects, or has a network link break, and there is no notice of this in the general case. (Even in the event of a crash and successful reboot, other SGs don't hear about it unless the rebooted SG has specific reason to talk to them immediately.) Over-quick response to temporary network out- ages is undesirable... but note that a tunnel can be torn down and then re-established without any user-visible effect except a pause in traffic, whereas if one end does reboot, the other end can't get datagrams to it at all (except via IKE) until the situation is noticed. So a bias toward quick response is appropriate, even at the cost of occasional false alarms.

A mechanism for recover after reboot is not specified in this

document, as it is a topic of current research.

A deliberate shutdown should include an attempt to notify all other SGs currently connected by phase 1 SAs, using Deletes, that communication is about to fail. (Again, these will be taken as teardowns; attempts by the other SGs to negotiate new tunnels as replacements should be ignored at this point.) And when possible, SGs should attempt to preserve information about currently-connected SGs in non-volatile storage, so that after a crash, an Initial-Contact can be sent to previous partners to indicate loss of all previously established connections.

10. Unresolved issues

10.1 Control of reverse DNS

The method of obtaining information is by reverse DNS lookup causes problems for people who can not control their reverse DNS bindings. This is an unresolved problem in this version, and is out of scope.

11. Examples

11.1 Plain-text usage (permit policy)

What follows are two example scenarios. The first is where Gate1 and Gate2 have always-cleartext policies, and the second where they have some OE policy.

```
-----------------------------------------------------------------------

  Alice         Gate1      DNS      Gate2          Bob
   (1)
    ------(2)--------------->
    <-----(3)---------------
   (4)----(5)----->----------(6)------>------(7)----->
    <----(10)-----<----------(9)------<------(8)------
   (11)----------->----------(12)----->-------------->
    <-------------<-------------------<---------------

             Figure 5: Timing of regular transaction

-----------------------------------------------------------------------
```

Alice wants to send a packet to Bob, say a ping packet. Without the presence of opportunistic encryptors, the following events occur:

(1) Human or application 'clicks' with a name

      (2) Application looks up name in DNS to get IP address

      (3) Resolver returns A record to application

      (4) Application starts a TCP session or UDP session, OS sends packet

      (5) Packet is seen at first gateway from Alice (SG-A) (Gate1 transitions through Empty connection to always-clear connection and instantiates a passthrough policy at the forwarding plane)

      (6) Packet is seen at last gateway before Bob   (SG-B)

      (7) First packet from Alice is seen by Bob

      (8) First return packet is sent by Bob

      (9) Packet is seen at Bob's gateway (Gate2 transitions through an Empty connection to always-clear connection and instantites a

passthrough policy at the forwarding plane)

      (10) Packet is seen at Alice's gateway

      (11) OS hands packet to application, Alice sends another packet

      (12) a second packet traverses the Internet

11.2 Opportunistic Encryption

In the presence of properly configured opportunistic encryptors, the event list is extended.

```
   ----------------------------------------------------------------------


     Alice          SG-A      DNS       SG-B          Bob
      (1)
       ------(2)-------------->
       <-----(3)---------------
      (4)----(5)----->+
                    ----(5B)->
                    <---(5C)--
                    ~~~~~~~~~~~~~(5D)~~~>
                    <------------(5E1)---
                    -------------(5E2)-->
                    <------------(5E3)---
                    #############(5E4)##>
```



```
                    <############(5E5)###
                             <----(5F1)--
                             -----(5F2)->
                    #############(5G1)##>
                             <----(5H1)--
                             -----(5H2)->
                    <############(5G2)###
                    #############(5G3)##>
                     ============(6)====>------(7)----->
       <-----(10)----<==========(9)=====<------(8)------
      (11)----------->==========(12)=====>-------------->
       <-------------<===================<---------------
```

Figure 6: Timing of Opportunistic Encryption transaction

```
   ----------------------------------------------------------------------
```

      (1) Human or application clicks with a name

      (2) Application initiates DNS mapping.

      (3) resolver returns A record to application

      (4) Application starts a TCP session or UDP

      (5) SG (host or SG) sees packet to target, buffers it.

      (5B) SG asks the DNS for TXT record

      (5C) DNS returns TXT record(s)

      (5D) Initial IKE Main Mode Packet goes out

      (5E) IKE ISAKMP phase 1 succeeds

      (5F) SG-B asks the DNS for TXT record to prove SG-A agent for Alice

      (5G) IKE phase 2 negotiation

      (5H) DNS lookup by responder (SG-B)

      (6) buffered packet is sent by SG-A

      (7) packet is received by SG-B, and decrypted, sent to Bob

      (8) Bob replies, packet is seen by SG-B

      (9) SG-B already has tunnel up with SG-A, uses it

      (10) SG-A decrypts packet and give it to Alice

      (11) Alice receives packet. Sends new packet to Bob

      (12) SG-A gets second packet, sees that tunnel is up, uses it

For the purposes of this section, we will describe on the changes that occur between Figure 5 and Figure 6. This means time points 5, 6, 7, 9 and 10.

11.2.1 (5) IPsec packet interception

At point (5), Security Gateway 1 intercepts the packet due to the lack of a policy (the non-existant policy!) for this source/destination pair. A hold policy is created, and the packet is buffered. A request is sent to the keying daemon for keys.

11.2.2 (5B) DNS lookup for TXT record

SG-A's IKE daemon, having looked up the source/destination in the connection class list, creates a new Potential OE connection instance. The DNS queries are started.

11.2.3 (5C) DNS returns TXT record(s)

DNS returns properly formed TXT delegation records, and SG-A's IKE daemon transitions this instance from Potential OE connection to Pending OE connection.

For the example above, the returned record might contain:

```
   ----------------------------------------------------------------------


   X-IPsec-Server(10)=192.1.1.5 AQMM...3s1Q==
```

Figure 7: Example of reverse delegation record

```
   ----------------------------------------------------------------------
```

with SG-B's IP address and public key listed.

11.2.4 (5D) Initial IKE Main Mode Packet goes out

Upon entering Pending OE connection, SG-A sends the initial ISAKMP message, with proposals, see Section 4.6.1.

11.2.5 (5E1) Message 2 of phase 1 exchange

SG-B receives the message. A new connection instance is created in the Unauthenticated OE Peer state.

11.2.6 (5E2) Message 3 of phase 1 exchange

SG-A sends a Diffie-Hellman exponent. This is an internal state of the keying daemon.

11.2.7 (5E3) Message 4 of phase 1 exchange

SG-B responds with a Diffie-Hellman exponent. This is an internal state of the keying protocol.

11.2.8 (5E4) Message 5 of phase 1 exchange

SG-A uses the phase 1 SA to send its identity under encryption. The

choice of identity is discussed in Section 4.6.1. This is an internal state of the keying protocol.

11.2.9 (5F1) Responder lookup of initiator key

Security Gateway 2 asks DNS for the public key of the initiator. This is done by asking for a KEY record by IP address in the reverse map. That is, a KEY resource record is queried for 4.1.1.192.in-addr.arpa (recall that SG-A's external address is 192.1.1.4) The resulting public key is used to authenticate the initiator. See Section 5.1 for further details.

11.2.10 (5F2) DNS replies with public key of initiator

Upon successfully authenticating the   peer, the connection instance is transitioned to Authenticated OE peer on SG-2.

The format of the TXT record that is returned is described in Section 5.2.

11.2.11 (5E5) Responder replies with ID and authentication

SG-B sends its ID along with authentication material. This is an internal state for the keying protocol.

11.2.12 (5G) IKE phase 2

11.2.12.1 (5G1) Initiator proposes tunnel

Having established mutually agreeable authentications (via KEY) and authorizations (via TXT), SG-A proposes to create an IPsec tunnel for datagrams transiting from Alice to Bob. This tunnel is established for only the A/B combination, not for any subnets that may be behind SG-A and SG-B.

11.2.12.2 (5H1) Responder determines initiator's authority

While the identity of the SG-A has been established, its authority to speak for Alice has not yet been confirmed. This is done by doing a reverse lookup on A's address for a TXT record.

Upon receiving this specific proposal, SG-2 transitions its instance into the Potential OE connection state. SG-2 may already have an instance, and the check is made as described above.

11.2.12.3 (5H2) DNS replies with TXT record

The returned key and IP address should match that of SG-A.

11.2.12.4 (5G2) Responder agrees to proposal

Should additional communication occur between, for instance, D and B via SG-A and SG-B, a new tunnel (phase 2 SA) would be established. The phase 1 SA may be reusable.

SG-A, having successfully keyed the tunnel, now transitions from Pending OE connection to Keyed OE connection.

11.2.12.5 (5G3) Final acknowledgement from initiator

The initiator agree with the responder's choice and sets up the tunnel.

The proper authorization returned with keys SG-2 to transition its instance to the Keyed OE connection.

11.2.13 (6) IPsec succeeds, sets up tunnel for communication between Alice and Bob

The packet which was saved at step (5) is sent through the newly created tunnel to SG-B. Bob receives it at (7) and replies at (8).

11.2.14 (9) SG-B already has tunnel up with G1, uses it

At (9), SG-B has already established an SPD entry mapping Bob->Alice via a tunnel, so this tunnel is simply applied. The packet is encrypted to SG-A, decrypted by SG-A and passed to Alice at (10).

12. Security Considerations

12.1 Configured vs Opportunistic Tunnels

Configured tunnels are those which are setup using bilateral mechanisms: exchanging public keys (raw RSA, DSA, PKIX), pre-shared secrets, or by referencing keys that are in known places (distinguished name from LDAP, DNS). These keys are then used to configure a specific tunnel.

A pre-configured tunnel may be on all the time, or may be keyed only when needed. The end points of the tunnel are not necessarily static: many mobile applications ("road warrior") are considered to be configured tunnels.

The key consideration is that configured tunnels are assigned specific security properties. They may be trusted in different ways. This is usually related to exceptions to firewall rules, exceptions to NAT processing, and to bandwidth or other quality of service restrictions.

Opportunistic tunnels are not inherently trusted in anyway. They are created without prior arrangement. As the two parties are strangers, there MUST be no confusion of datagrams that arrive from opportunistic peers and those that arrive from configured tunnels. A security gateway MUST take care that an opportunistic peer can not impersonate a configured peer.

It is recommended that an implementation MUST prevent opportunistic peer from impersonating another opportunistic peer. This is equivalent to inforcing ingress filtering of source addresses.

An implementation suggestion is to logically treat opportunistically tunnel datagrams as if they arrive on a distinct logical interface

from other configured tunnels. As the number of opportunistic tunnels that may be created automatically on a system is potentially very high, careful attention to scaling should be taken into account.

As with any IKE negotiation, opportunistic encryption cannot be secure without authentication. Opportunistic encryption relies on DNS for its authentication information and therefore cannot be fully secure without a secure DNS. Without secure DNS, it can protect against passive eavesdropping, but not against active man-in-the-middle attacks.

12.2 Firewalls vs Opportunistic Tunnels

Typical usage of per-packet access control lists is to implement

various kinds of security gateways. These are typically called "firewalls".

Typical usage of virtual private network (VPN) within a firewall is to bypass all or part of the access controls between two networks. Additional trust (as outlined in the previous section) is given to datagrams that arrive in the VPN.

Datagrams that arrive via opportunistically configured tunnels MUST not be trusted. Any security policy that would apply to a packet arriving in the clear SHOULD also be applied to datagrams arriving opportunistically.

12.3 Denial of Service

There are several different forms of denial of service which an implementor should concern themselves with. Most of these problems are shared with security gateways that have large numbers of mobile peers (road warriors).

The design of ISAKMP/IKE, and its use of cookies defend against many kinds of denial of service. There is an assumption that if the phase 1 (ISAKMP) SA is authenticated, that it was worthwhile creating. Opportunism changes this assumption. As the gateway will communicate with anyone, it is possible to form phase 1 SAs with any machine on the Internet.

13. IANA Considerations

There are no known numbers which IANA will need to manage.

14. Acknowledgements

Thanks to Tero Kivinen, Sandy Harris, Wes Hardarker, Robert Moskowitz, Jakob Schlyter, Bill Sommerfeld, John Gilmore and John Denker for their comments and constructive criticism.

References

[1] Redelmeier, D. and H. Spencer, "Opportunistic Encryption", paper http://www.freeswan.org/freeswan_trees/freeswan-1.91/doc/opportunism.spec, May 2001.

[2] Defense Advanced Research Projects Agency (DARPA), Information Processing Techniques Office and , "Internet Protocol", STD 5, RFC 791, September 1981.

[3] Braden, R. and J. Postel, "Requirements for Internet gateways", RFC 1009, June 1987.

[4] IAB, IESG, Carpenter, B. and F. Baker, "IAB and IESG Statement on Cryptographic Technology and the Internet", RFC 1984, August 1996.

[5] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

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

[7] Piper, D., "The Internet IP Security Domain of Interpretation for ISAKMP", RFC 2407, November 1998.

[8] Maughan, D., Schneider, M. and M. Schertler, "Internet Security Association and Key Management Protocol (ISAKMP)", RFC 2408, November 1998.

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

[10] Huttunen, A. and W. Dixon, "UDP Encapsulation of IPsec Packets", ID internet-draft (draft-ietf-ipsec-udp-encaps-00), June 2001.

[11] Kivinen, T. and M. Kojo, "More MODP Diffie-Hellman groups for

IKE", ID internet-draft (draft-ietf-ipsec-ike-modp-groups-03), November 2001.

[12] Vixie, P., "Extension Mechanisms for DNS (EDNS0)", RFC 2671, August 1999.

[13] Rosenbaum, R., "Using the Domain Name System To Store Arbitrary String Attributes", RFC 1464, May 1993.

[14] Eastlake, D., "Domain Name System Security Extensions", RFC

2535, March 1999.

[15] Eastlake, D., "RSA/MD5 KEYs and SIGs in the Domain Name System (DNS)", RFC 2537, March 1999.

[16] Eastlake, D. and O. Gudmundsson, "Storing Certificates in the Domain Name System (DNS)", RFC 2538, March 1999.

[17] Durham, D., Boyle, J., Cohen, R., Herzog, S., Rajan, R. and A. Sastry, "The COPS (Common Open Policy Service) Protocol", RFC 2748, January 2000.

[18] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.

Authors' Addresses

Michael C. Richardson
Sandelman Software Works
470 Dawson Avenue
Ottawa, ON K1Z 5V7
CA

EMail: mcr@sandelman.ottawa.on.ca
URI: http://www.sandelman.ottawa.on.ca/

D. Hugh Redelmeier
Mimosa
Toronto, ON
CA

EMail: hugh@mimosa.com

Henry Spencer
SP Systems
Box 280 Station A

Toronto, ON M5W 1B2
Canada

EMail: henry@spsystems.net

Full Copyright Statement

Acknowledgement

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