Network Working Group   IPsec Working Group
INTERNET DRAFT   C. Madson
Expires: May 2001   Cisco Systems Inc.
                                                             November 2001

Protocol Requirements for Son-of-IKE

<draft-ietf-ipsec-son-of-ike-protocol-reqts-00.txt>

Status of this Memo

Abstract

Various proposals have been made for updating the IKE protocol.

One thing which is missing from the dicussion is an evaluation of the scope of IKE, identifying which problems it should solve. Once this scoping is done, it becomes easier to specify the requirements for the protocol, as well as the protocol itself. Sections of this document discuss various scenarios that are considered important for IKE; this list needs to be refined by the WG.

This document also makes recommendations for protocol improvement in such areas as modularity, extensibility, protocol convergence and simplicity, which are important regardless of the scope of IKE.

1. Introduction

While there have been various proposals made over time of how to "improve" IKE, in order to address issues such as complexity, it's important to decide what are the important characteristics for the protocol, and then determine what changes need to be made in order to accomplish this.

Another way of stating this is "what are the characteristics of an optimal protocol, and what does it take to get there?" It will be important to prioritize these characteristics, in order to help focus on making changes in areas that will have the biggest benefit.

This document does not disucss security requirements in the sense that it does not cover things like DoS, replay protection, cryptographic protection, etc. It concentrates more on the protocol aspects of IKE -- in general these concepts could be applied to other non-security protocols.

2. Conventions Used in This Document

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 [1].

3. A Need for Protocol Scoping

While the WG considers what should be changed within IKE, one of the early questions that should be answered is one of "what should be the scope of IKE?" This issue has several different facets.

There has been this (at least implicit) general philosophy that if any other protocols need security, that the security needs to be provided by IPsec, and by proxy, the key management needs to be provided by IKE.

While IKE should be the key management mechanism of choice for the vast majority of usage domains, requiring any mechanism to "solve the world's problems" is not only inappropriate, but results in a protocol that is difficult to work with.

There are several examples in the IETF where the lack of a well-defined scope results in protocols and mechanisms that are either overly generalized (and therefore harder to evaluate and understand), or where various parties attempt to shoehorn in the "little piece that will solve their problem". Any protocol/mechanism forced to grow such appendages in abnormal places over time becomes unrecognizable. Or, to restate it in a more straightforward way, even if the mechanism started out simple, these little pieces shoved into such a mechanism can over time vastly increase the complexity of the mechanism. Such a protocol can become cursed by its own success.

There needs to be a description of just what IKE will (and won't) do. The WG also needs to decide upon the important environments in which

IKE will be used, in order to ensure that IKE meets the needs to those environments. In some cases, there are pieces needed that aren't currently supplied by IKE; a conscious decision needs to be made whether or not to accommodate that piece within IKE, or to leave it to an external mechanism. [In the latter case, if that mechanism does not exist, a statement of requirements associated with that mechanism must be written.]

While a well-defined scope is very important, this does not mean that IKE should not be considered as a solution for other usage domains. However, while there is an understandable desire to use IKE everywhere, it is appropriate to consider how much would need to be added/changed to IKE, and somehow evaluate just "how much is too much?" In certain cases, the answer may be to add something to IKE, while in other cases a separate protocol may be needed, to be used either in conjunction with IKE or instead of IKE.

While it may not always be appropriate to use IKE in its normal form, that doesn't mean that IKE needs to be jettisoned altogether. To a large extent, many of the things provided by IKE are also things that are needed regardless of the mechanism used.

One such current example is KINK, which has used a lot of IKE components, while replacing the key exchange and authentication components. The WG evaluated the various components of IKE, and decided which pieces it could reuse from IKE and which it had to replace. Another current example is multicast, where a key management mechanism that needs to address such issues as delivering keying material to a group of "valid" members is also reusing various components of IKE.

There has been this long-standing view that at least ISAKMP can be used to secure things other than IPsec (and possibly the encapsulation protocols coming out of MSEC). In fact, this represents one of the fundamental design criteria applied to ISAKMP, via the use of Domains of Interpretation. The vision of the Domains of Interpretation was to allow other entities to use the vast bulk of the ISAKMP protocol, while simply filling in a few pieces that are protocol-specific. In other words, the protocol payloads, exchanges, etc., are the same, but some of the payload contents could be unique to the particular DOI.

There have been those that have argued that the concept of Domains of Interpretation has overly complicated the protocol, in part because it forces the use of multiple documents to describe the protocol for a particular DOI. On the other hand, it currently represents the only "official" mechanism for specifying IKE variants that are actually very close to IKE in many ways.

There have been several short-lived IETF-based proposals made in the past attempting to use the DOI concept. Today there are two current proposals: one for the multicast keying mechanism, known as "Group

DOI" (GDOI) coming out of MSEC, and another is the MAP Security Domain of Interpretation (MAPSEC DOI), which instantiates ISAKMP for use with MAP.

Another area of scooping has to do with "what percentage of the problem should be solved by IKE and what should be solved by an external mechanism?" While IKE provides a nice protected pipe to pass information between the two crypto peers, it shouldn't be viewed as a license to cram everything into there. Nonetheless, in some cases the best answer may be to have the information passed through IKE. [The current proposal associated with XAUTH and MODECFG uses IKE to pass protocol information which is needed prior to IKE phase 2 processing.]

Many of the above issues need to be considered when deciding the appropriate scooping for IKE. What is also important is that the WG needs to consider some number critical domains of usage that must be addressed within the base protocol; in other words, which subset of the "world's problems" are important to be tacked by IKE.

As an initial proposal, the base IKE protocol should address the following scenarios: (a) end-to-end security, (b) secure remote access for clients connecting to a device that will subsequently enable them to access the networks behind it, and (c) VPN site-to-site tunnels. The appendix will discuss the requirements for these scenarios.

It will also become very important to consider protocol modularity: as the WG decides what should be considered "IKE", it will be important to develop an approach towards modularity that can provide a means for others to use components of IKE within their own protocols. A well-thought out strategy for extensibility will also be important. Input and output criteria for each component of IKE must be carefully described.

4. **Ease of Extensibility**

Extensibility becomes a critical characteristic: if it becomes too difficult to extend a protocol, the protocol is as good as dead in the face of ever-changing network topologies. [This does not mean that the solution to all the problems is to simply extend the protocol - in certain cases the more appropriate answer is to use pieces of IKE where appropriate, as discussed in Section 3.]

Via the use of IANA-controlled numbering spaces, it is possible to specify new field values (e.g. new algorithms) or even new payloads or exchanges. There are a few obvious issues with the current IKE as outlined below:

1. The biggest problem here is that the protocol (or at least most

implementations of it) will not only reject unrecognized payloads, but will also force a termination of the connection. In certain cases, this behavior is too drastic, especially for payloads that could be considered optional and probably not security-critical.

When there is a strong need to use a payload that did not exist in the original IKE specification, the parties are forced to use Vendor-ID payloads to either pre-negotiate support for such a payload, or to actually encode the payload itself. This works because most vendors will ignore unrecognized Vendor-ID payloads.

Forcing the use of such mechanisms results in unnecessary complexity. The new IKE specification must be able to at least represent payloads in such a way that the receiver can clearly understand whether or not it is valid to ignore a particular payload that it doesn't recognize. In other words, if a payload is "critical", it will be represented in a way that the receiver will reject it if it doesn't understand it (and possibly terminate the connection). If the payload is "non-critical", it will be represented in such a way that the receiver can ignore it. When new payloads are defined, they will be required to state whether they are always critical, critical by choice of sender, or non-critical.

2. IKE currently only hashes part of the packet. The algorithm to decide which fields to hash does not accommodate new payloads, regardless of their security criticality. The new IKE specification must address this deficiency.

3. An analysis needs to be done with respect to other fields, in order to decide the appropriate behavior for handling unrecognized values. This should be spelled out in the new IKE specification.

Besides these issues associated with extensibility of fields and values, extensibility also needs to be considered along with modularity in order to address some of the issues discussed in Section 3. A good definition of modularity can help in terms of thinking about extensibility, and vice-versa.

One such example is the use of the DOI as an extension point - depending upon how protocol modularity is approached, a decision can be better made as to whether to retain the DOI concept or to replace it with something different that accomplishes the same thing.

Another area of extensibility has to do with the general idea of "now that there is a protected (IKE) connection between the two crypto endpoints, what else besides phase 2 negotiation can be protected via this connection?" [This could also apply to new exchanges happening during phase 1, or even before phase 1.] If these new messages or exchanges have an effect on other parts of the protocol, such as a phase 2 negotiation, or even introducing a new phase to the protocol, the specification of the new message/exchange must analyze and document this.

5. Modularity

Protocol modularity can be helpful in several ways: it can help sub-divide the protocol itself into components that are more easily grasped and more easily analyzable.

Protocol modularity can also help to make "modules" available for use by other key management mechanisms, in the case where it is determined the IKE is not the best fit, or to allow other mechanism to use "most" of IKE and only replacing the one or two modules necessary for supporting that mechanism's requirements.

The effort of sub-dividing the protocol needs to include some definition of the boundaries, e.g. what does this module require from other modules, and what will be produced by this module? In general, a module can be replaced by another module as long as that second module meets the needs that were supplied by that first module. [A high-level illustrative example could be "anything could replace IKE as long as that new mechanism(s) met the needs of IPsec, which include negotiating SAs and supplying bulk keying material.]

One example of a module might be key agreement; given the security importance of key agreement, having good modularity would allow the key agreement piece to be replaced if, for example, the original mechanism were deemed to be no longer secure. [Or, certain environments could call out for even stronger key agreement mechanisms that are too processing-intensive for normal use. Or other environments may wish to replace the key agreement with another mechanism based on base key material being generated by a trusted third party.]

If key agreement were its own module, it could be evaluated in isolation. The evaluation could place some criteria on the module inputs (randomness requirements, etc.), and try to make a statement as to the amount of security provided by this module. It could also make some statements about processing impact (time/number of messages, etc.). Such information could help to decide whether to use module A or module B for a particular environment.

6. Improve Convergence Characteristics

It is important for a protocol to behave in a predictable manner, even in the face of retransmissions, lost packets, receipt of invalid packets or payloads, loss of communication between the peers, etc.

It is especially important that once an IKE exchange starts, that the two peers have a very similar view of the state of the connection between them.

It is also important for the behaviors to be well understood, so that implementers can make sure that they have implemented the protocol correctly.

This can be addressed by various means. While representations such as protocol state machines are always useful, it is especially important to analyze error conditions and to document the analysis.

The base protocol must specify all of the pieces needed to ensure convergence. For example, there have been various mechanisms suggested over time to address such things as detecting loss of communication with the IKE peer, and how to handle synchronization between the IKE peers in terms of rekeying. While in some cases the suggested mechanism might be the best solution to the problem, in some cases, the mechanism has been introduced in order to address a deficiency in the base protocol (e.g. lack of guaranteed notify/delete messages). The eventual solution may involve the merging of both the new mechanism and other protocol changes.

In any case, the base protocol specification must be expanded to address the following issues, which affect the ability of the two peers to have the same view of the connection:

* Rekeying: the behaviors associated with rekeying must be spelled out

* Detection of loss of communication with peer

* Documenting errors, especially if it results in a state change to a peer. Especially if it involves sending an error message to the peer. Besides documenting the error, there needs to be documentation describing what a peer needs to do when receiving the error message. Defining a list of well-known errors and well-known behaviors relative to the errors can help to ensure more consistent behavior.

* Negotiation matching rules need to be spelled out (e.g. SA lifetimes, etc.)

As new things (exchanges, etc.) are introduced to the protocol, these specifications will also be required to perform a similar analysis. This becomes especially important if the addition has an impact on protocol behaviors.

7. Improve Simplicity

Simplicity has various facets to it. The primary goal is to make a protocol that is understandable and easier to implement.

One approach to simplicity is getting rid of unnecessary fields, exchanges, etc. and reducing the number of documents needed to describe the protocol.

Good scoping and modularity can also help to make a protocol easier to understand. Good descriptions of protocol behaviors can also make the protocol easier to understand and implement.

Another facet of simplicity has to do with "ease of accomplishing a particular function". One example is the area of negotiation - while having one peer simply send a list of proposals to the other peer, and having the receiver simply pick one is "simple" on the one hand, if the goal is "connectivity even without a-priori complete knowledge of the policy configured on the receiver", things get complicated quickly. If the negotiation only included two attributes, it would be (relatively) easy for the initiator to derive a complete list of possibilities. When there are more than two attributes, and some of the attributes are variables such as lifetime, things becomes complicated quickly.

In some cases, such as the negotiation one above, it may be that the "simplicity" will be accomplished by the use of something external to IKE (e.g. policy discovery mechanism, etc.). Such assumptions and decisions must be documented.

8. Authentication and IKE

There has been this commonly held view that "IKE requires an X.509 PKI", possibly because the use of X.509 certificates and pre-shared keys are the two main authentication mechanisms spelled out in the base draft (at least to the point of specifying payloads and the use of information derived from the authentication to generate the SKEYID). In some cases, this perception results in people not wanting to consider using IKE. Others simply feel that such a specification should not be a part of the IKE document, that such an inclusion discourages the development/deployment/use of other mechanisms.

One solution is to remove the specifics to separate drafts (except for pre-shared keys), while the base document spells out requirements for authentication. Then separate documents could be written for the protocol and handling requirements for each authentication mechanism (e.g. "if you want to do X.509 certificates, you must support these payloads, you must support these encoding formats, you must send these payloads at this point in the exchange, you must send certificate chains that look like this, you must be able to support certificate path discovery (or not), you must feed this blob into SKEYID generation this way, etc..."). The document should also spell out if the particular authentication mechanism implies a particular trust model.

Over time, there has also been confusion with respect to user-vs.-machine authentication via IKE. Over time, it seems that the general consensus has been that IKE provides machine authentication; part of it has to do with the wide variety of user authentication mechanisms, and where or not they should be accommodated for IKE phase 1, especially given that IKE is typically constrained to a 6-message exchange. If this model holds, there may be a need for user authentication as a part of IKE (and not just in the remote access scenario). At a minimum, the base draft should specify that IKE supplies machine-level authentication.

9. Security Considerations

This document describes various considerations that should be taken into account for developing a follow-on protocol to IKE. The primary goals of these considerations is to provide guidance for a protocol that is more easily understood/evaluated in terms of security properties and behaviors.

10. Acknowledgements

Thanks to various members of the IPsec WG whose comments over a long period of time have contributed to the thinking which resulted in this document. Thanks also to Brian Weis for reviewing this document and making many helpful suggestions.

11. References

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

12. Editor's Address

Cheryl Madson
<cmadson@cisco.com>
Cisco Systems, Inc.

The IPsec working group can be contacted through the chairs:

Barbara Fraser
<byfraser@cisco.com>
Cisco Systems, Inc.

Ted T'so
<tytso@mit.edu>
Massachusetts Institute of Technology

A. Scoping: Primary Domains of Usage

As was discussed earlier, it is important for IKE to focus on some number of usage domains, and make sure that the domains' requirements have been met. The following sections describe several characteristics of each domain of usage which are noteworthy.

The scenarios listed here are considered to be the major ones that should be addressed, although the list is not exhaustive and needs to be refined by the WG.

Many times, IP traffic between a given source and destination will be secured via one of these scenarios; however, many of these can be nested (e.g. a client has to run the remote access scenario to get into the corporate network, and within that IPsec-protected connection it can run another IPsec connection between the client and the intended server.

[Note: In this version of the draft, the information on these scenarios is intentionally sparse. While this tries to call out key characteristics, further analysis is needed to determine if the categories are sufficient. Need to decide how SCTP should be integrated with these scenarios. May need discussion of QoS in certain scenarios.]

A.1. Virtual Private Network Site-to-Site Tunnels

* Operational Characteristics: VPN Site-to-Site Tunnels are between two devices acting as Security Gateways (SGWs). IPsec can either directly encapsulate the IP traffic that it will secure, or it can secure another tunneling protocol (e.g. IPinIP) running between the two SGWs that has already encapsulated the data.

A pair of SGWs can support several simultaneous IPsec tunnels in between them. These tunnels can have the same or different protection.

* Policy: In many cases, in order to ensure common policy on the SGWs, the SGWs need to be configured/provisioned. However, policy discovery mechanisms are also desirable.

* NAT: The IPsec connection may be NAT'd between the SGWs. The traffic inside the IPsec connection may also be NAT'd by one of the SGWs.

* Dynamic Addresses: In many cases one of the SGWs may have a dynamically-assigned address. If both SGWs have dynamically-assigned addresses, at least one requires a fixed DNS name where the DNS entry is updated appropriately.

* Authentication: Machine-level only.

A.2. Secure Remote Access

* Operational Characteristics: When the client is outside the protected network that it needs to access, it will have to interact with a Remote Access Server (RAS) in order to access that network.

In many cases, the Secure Remote Access scenario is replacing a remote access scenario that uses dial-up access to reach a network. The client would dial up to the Remote Access Server (RAS) and interact with the RAS to perform user-level authentication (e.g.

userid/password, SecureID, etc.). If the authentication is successful, the RAS would permit access to that network, and may also push down configuration information to the client.

In the Secure Remote Access case, before the IPsec tunnel is constructed between the client and the RAS, the client must first authenticate itself (most likely using legacy authentication mechanisms). The IPsec connection will most likely operate in tunnel mode.

The entity that supplies the RAS typically requires a great deal of accounting of connections and resources consumed, in order to do capacity planning.

* Policy: In many scenarios, there is a need to download policy information from the RAS to the client. [Note: are both policy pull (client request) and policy push (server push to client) models supported?]

* NAT: The outside IPsec connection may be NAT'd between the client and the SGW. If an internal address is assigned to the client, there should be no need for the RAS to perform NAT translation on the traffic that is sent/received in the IPsec connection.

* Dynamic Addresses: The client will have a dynamically assigned outside IP address assigned from its local POP. In many cases it will get an internal address assigned by the RAS so that the RAS will not have to NAT translate the traffic. The client could request this address, or the RAS could push it down along with other configuration information needed by the client (info on how to reach the inside DNS server, etc.).

The RAS will typically have a static IP address (or at least a static DNS name, where the DNS entry is updated appropriately).

* Authentication: In many cases (e.g. the Internet Kiosk), machine-level authentication becomes meaningless. In other cases (e.g., a remote laptop) both machine-level authentication and user-level authentication are valuable. However, there is a strong requirement for user-level authentication, requiring some amount of user interaction. The user- level authentication is mainly done via legacy authentication mechanisms (e.g. SecureID, userid/password, etc.).

A.3. End-to-End Security

What this scenario attempts to describe is the characteristics associated with the common end-to-end scenarios. The primary definition of end-to-end here is that the traffic sourced by/destined to a pair of nodes is secured by those nodes.

The most common "end-to-end" scenario is an IPsec connection between two workstations. "End-to-end" can also apply to securing a network management path between the management station and a router.

However the "end-to-end" concept can be applied to a lot of different scenarios that may have additional requirements. For example, securing OSPF traffic between OSPF neighbors requires a system to share a key amongst its neighbors on a link, as it doesn't explicitly learn about its neighbors until it runs the OSPF "hello" protocol. Scenarios with special requirements should be called out separately.

* Operational Characteristics: An IPsec "connection" is used to secure traffic between the two IP endpoints for that traffic. The granularity of the connection may be an individual socket, or all traffic between the two systems. If multiple connections exist, they can have the same or different protections. The IPsec connection can operate in either tunnel or transport mode.

* Policy: In some cases this is pre-configured/provisioned. However, policy discovery mechanisms are desirable here.

* NAT: The IPsec connection may undergo NAT translation. The traffic within the IPsec connection most likely will not be NAT'd.

* Dynamic Addresses: It's possible for one node to have a dynamic address. In most cases the other node will have a fixed address, or at least a fixed DNS name, where the DNS entry is updated appropriately.

* Authentication: In many cases, machine-level authentication is desirable and sufficient.

In some cases, such as client-to-server connections, user-level authentication may also be necessary. User-level authentication via legacy mechanisms is currently outside the scope of IKE. However, it may be desirable to have such user authentication happen before IPsec connections could be established, similar to what happens in the remote access scenario. User-level authentication can also be done via user credentials that can be easily passed in IKE phase 1 (e.g. X.509 user certificates).

A.4. Mobile IP

[No doubt there are gaps in this particular scenario...]

* Operational Characteristics: IPsec is used to secure a Mobile IP tunnel (which has its own encapsulation). In some ways this scenario is like a remote access scenario. The IPsec connection can operate in either tunnel or transport mode. IPsec is also used to secure the Binding Update messages.

* Policy: Most likely pre-configured/provisioned while the mobile

user is on the home network.

* NAT: The IPsec connection may be NAT'd. The traffic carried by the IPsec connection most likely will not be NAT'd.

* Dynamic Addresses: The mobile node will have a dynamic address. In fact, it will have several different ones over a period of time.

* Authentication: Depending upon the environment, the mobile device may be required to authenticate to a local node that gives it permission to access the local media; such authentication is outside the scope of IKE/IPsec.

In many environments, machine authentication for IKE will be sufficient, in that the user must possess a mobile node device that has the appropriate credentials already pre-installed as a part of initial provisioning.