IPSEC Working Group
INTERNET-DRAFT
draft-ietf-ipsec-ikev2-rationale-00.txt
Expires: August 2002

Dan Harkins
Charlie Kaufman
Tero Kivinen
Stephen Kent
Radia Perlman
February 2002

Design Rationale for IKEv2
<draft-ietf-ipsec-ikev2-rationale-00.txt>

Status of this Memo

Abstract

This document describes the reasons for the design choices in IKEv2, the protocol described in draft-ietf-ipsec-ikev2-01.txt. This document describes why certain features are supported, and explains the modifications between the second draft of IKEv2 and the first. It describes both the changes we chose to make and the changes that we considered but chose not to make. The changes are minor and mostly based on feedback received from the first draft.

1. Introduction

IKEv2 [Har01] has been presented to the IPsec WG as a possible successor to IKE (IKE is documented in RFCs 2407, 2408, and 2409). IKEv2 preserves most of the features of the original IKE, including

identity hiding, perfect forward secrecy, two phases, and cryptographic negotiation, while greatly redesigning the protocol for efficiency, security, robustness, and flexibility. This document explains the reasoning for the design choices made by IKEv2, as well as possible alternatives, the advantages and disadvantages of these alternatives, and a description of how the alternatives, if desired, would be implemented.

2.0 Features for an IKE successor

At a minimum, such a protocol should perform mutual authentication between two parties and establish cryptographic keys for integrity and encryption of an IPsec SA. Simplicity is always a goal in any protocol. Additional features, apropos a key and security association management protocol for IPsec, include identity hiding, cheap and graceful rekeying, dead peer detection, plausible deniability, support for multiple services co-located at an IP address, negotiation of peer-dependent IPsec policies, and a two-phase structure to enable inexpensive creation of multiple SAs between the same two hosts or security gateways. IKEv2 is a major redesign of IKEv1. Syntax is preserved when there is no compelling technical reason to change it, so that there might be some ability to preserve code. However, IKEv2 is not (backwards) compatible with IKEv1. A node that implements both IKEv2 and IKEv1 can interwork with an IKEv1 node by detecting that the peer implements only IKEv1, and thereafter communicating using only IKEv1.

2.1 Why Two Phases?

In IKEv2 terminology, the first phase is known as the IKE-SA. Once the IKE-SA is created, it can be used for sending authenticated notification messages, reliable dead-peer detection, and inexpensive creation of "child-SAs", which are IPsec SAs for subscriber traffic. In theory, IKE could facilitate creation of SAs for other protocols as well.

We argued in [PK01] that having two phases was unnecessary and added complexity. Since then, we have gotten feedback that the two phase structure is useful and in typical IPsec deployments, multiple child-SAs are indeed created. Also, once the details of the design were worked out, we found that the ability to send authenticated notification messages made certain necessary features, such as dead peer detection, simpler and more robust.

Why do people find it useful to create multiple IPsec SAs between the same pair of hosts?

- avoiding multiplexing multiple conversations over the same SA.

Several years ago Bellovin pointed out that if encryption is done without integrity protection, there is a splicing attack whereby a process involved in one flow through a shared crypto implementation can, through an active attack, cause traffic for a different flow to be decrypted and delivered to the process in the first flow. Of course, nobody should be doing encryption without integrity protection under these circumstances. It is likely there is no similar flaw if integrity is used. But in a case where a security gateway is delivering traffic on behalf of multiple customers, and the data is going to another security gateway in order to access other machines of those customers, the customers feel safer knowing that their traffic is being carried via a different SA (and different key) than traffic between nodes belonging to other customers.

- different security properties of different flows. According to policy, some traffic might be only integrity-protected. Other traffic might be encrypted with a short key. Other traffic might be encrypted with a long key. Other traffic might use a vanity crypto algorithm designed by one of your customers, and it will make them happy if you use their algorithm for their traffic. In [PK01] we argued that all traffic might as well be protected according to the needs of the traffic that requires the strongest protection, but there might be performance reasons or legal reasons (or vanity reasons) why this is undesirable.

- different SAs for different classes of service. There might be different classes of service, such as priority classes, that might cause traffic for one class to travel much more slowly to the SA destination than other types of traffic to the same SA destination. To avoid replay attacks, the recipient keeps track of which sequence numbers have been received. Typically, it only keeps track of the highest n sequence numbers, up to the highest sequence number it has seen on this SA, and data with sequence numbers lower than the "left" side of the receive window are discarded. If different classes of service have widely disparate delivery times, the recipient would have to maintain a (potentially much) larger receive window to avoid inappropriate discarding of "late arriving" IPsec traffic.

In addition to these reasons for creating multiple child-SAs, we found that IKEv2's robustness is enhanced by its ability to use the IKE-SA to send reliable and authenticated informational messages. This detects error conditions, allows rekeying, and provides a method for detecting a dead peer.

2.2 Identity Hiding

Some people argue that identity hiding is an exotic feature that cryptographers put into IKEv1 just because they could. In many cases,

such as those where nodes are at fixed IP addresses, the identity is not hidden.

And, there are different flavors of identity hiding. IKEv2 provides identity hiding for both parties from passive attackers. Beyond that, one might want to hide the identities from active attackers. With public encryption keys, if at least one side already knows the identity and public key of the other, then it is possible to protect both sides from any active attacker (assuming the encryption key is not escrowed or otherwise compromised). With pre-shared secret keys, assuming both parties already know who they expect to be speaking with (within a small set, perhaps), it is also possible to protect both identities from active attack.

However, with public signature keys, one side or the other has to reveal its identity first. If it is talking to an active attacker, it will have revealed its identity. In [PK01] we argued that it was more important to protect the identity of the initiator, since in the client-server model, the server would be at a fixed IP address and would not have a hideable identity. However, we later realized that a much easier attack is a polling attack, in which the attacker merely opens an IPsec connection to a node. If the responder reveals its identity first, then this simple attack, which is easier to mount than a passive attack, will reveal the identity at that address. If the model were changed to a strict client-server model in which clients never respond to connections, and server identities are not important to protect, then it is reasonable to have the responder reveal its identity first. The feedback we got from the WG was that they did not want to change IPsec from a peer-ti-peer protocol into a client-serverprotocol.

To avoid a polling attack, (in which an active attacker simply initiates a connection to an IP address to find the identity associated with that IP address) IKEv2 has the initiator reveal its identity first. The active attack that IKEv2 has chosen not to deal with involves having someone impersonate Bob's IP address and discover the identities of parties that attempt to communicate with that IP address. This attack is more difficult to mount (than the polling attack described above) and it is not obvious what benefit it would provide the active attacker. Alice has only initiated a connection to an IP address. If she is not speaking to the real Bob, she will discover this and break the connection. So the active attacker cannot prove she intended to speak to Bob; merely that she initiated an IPsec connection to a particular IP address.

2.3 Plausible Deniability

This feature enables Alice and Bob to communicate, leaving no ability

for anyone to prove that they did have a conversation, even with the collusion of one of the parties. Alice does indeed sign a message proving she is Alice, but she does not sign Bob's name or IP address. The only thing she signs that belongs to Bob is his nonce, but anyone could have generated that nonce. It is sent in the clear, and so the fact that Bob generated that nonce certainly does not imply that nobody else could have generated that nonce. The fact that Alice carried on an IPsec connection with someone that used that nonce does not prove she talked to Bob.

In the first draft of IKEv2, we had Alice sign the entire contents of messages 1 and 2. This meant that all fields would be integrity protected, either through Alice's signature (and Bob also signed both messages), or through the integrity protection that all packets beyond the first two automatically had through IKE's integrity protection. Sara Bitan suggested a modification so that each side only signs the fields it generates, plus the other side's nonce. This still protects all fields, either through Bob's signature or Alice's. It also makes it more obvious that the protocol provides plausiable deniability. Some people argue that having Alice additionally sign Bob's Diffie-Hellman value proves she talked to Bob, because only Bob can produce the private exponent associated with that Diffie-Hellman number. However, all it proves is that Alice talked to someone who presented that Diffie-Hellman number. The fact that Bob knows the private exponent does not mean that it was Bob. Instead, it could be someone else that told Bob the Diffie-Hellman parameters.

2.4 Cookies

Although in [PK01] we explained how to effect a stateless cookie exchange without adding messages (by having Alice repeat, in message 3, what she said in message 1, in IKEv2 we chose to have Bob request stateless operation with an extra round trip, so the exchange in this case would be 6 messages. The reasons for this:

- Having the message be 4 exchanges does not come for free. Repeating everything in message 3 makes message 3 longer.

- Message 3 is likely to be large (it can contain Alice's certificate). If message 3 needs to be fragmented, then Bob cannot be stateless. He has to reassemble message 3 before he can determine if the cookie is correct. Therefore, it was simpler to ensure that all the messages before Bob starts needing to keep state are sufficiently small as not to require fragmentation.

2.5 Cryptographic Negotiation

In IKEv2, Alice proposes a set of potential cryptographic algorithms,

and Bob chooses. Why is it necessary to negotiate? Why not let one side or the other choose? Suppose IPsec just agreed upon a particular set of cryptographic algorithms, and, say, Bob told Alice what cryptographic algorithms to use. The problem with this is that if the world ever wanted to phase in a new cryptographic algorithm, there would be no way to do it until Bob could be assured that all the clients had upgraded to support it.

In SSL, cryptographic negotiation is based on complete suites rather than, as in IKEv1, individually negotiating each cryptographic algorithm. In the first draft of IKEv2 we cleaned up the syntax somewhat to avoid exponential explosion if many of one type of algorithm can work with any of a number of choices for some other type of algorithm. We argued on the list and during the meeting for moving towards suites.

We considered switching over to suites in this draft of IKEv2, but we got a surprising amount of feedback that people prefer negotiating individual algorithms. Since there was no compelling reason to change, and most opinions we heard were in favor of individual algorithm negotiation, we kept it the way it was. If there were working group consensus to move to suites, we would be delighted to make the change.

3.0 Changes in this version of the draft

Very little needed to be changed from the previous draft of IKEv2. In this section we describe the changes we considered making, but decided against, and the changes we did decide to make.

3.1 Changes we considered and rejected

We considered making the following changes, but decided against them:

- negotiating cryptography with suites rather than individual algorithms

- changing the encoding to "type of this payload" rather than "type of next payload". We find the "type of next payload" harder to understand, and if we had been laying out the packet formats from scratch, we would have argued for that encoding. However, there is no compelling reason in terms of compactness or parsing efficiency to change it, and one of our stated goals in IKEv2 was not to make "gratuitous changes" so that implementers might be able to reuse code. This change would have been totally a subjective preference.

- changing to an exchange that allows Bob to have stateless operation and still have the exchange fit within 4 messages. As we explained

above, we decided the exchange was better the way it is; 4 messages with an extra round trip if Bob wants Alice to return his stateless cookie before the exchange begins in earnest. It seemed like fitting within 4 messages in this case was only motivated by "bragging rights", and the downside was making message 3 bigger all the time, and the issue that if message 3 is sufficiently large to need fragmentation, then Bob would not be able to do stateless operation.

- adding in Bill Sommerfeld's "birth certificate" idea. In this idea Bob keeps a number in nonvolatile memory that increments each time the node restarts. When Bob restarts, he signs a "birth certificate" stating what the value of that counter is. This birth certificate is transmitted as a payload in message 4. Alice keeps this value. If Bob ever receives an ESP packet that doesn't decrypt properly or with an unknown SPI, he responds to that packet with his birth certificate. If the recipient has an SA for Bob with an older birth certificate, this lets them know Bob has restarted and forgotten state for that SA. We decided not to add that to this version of the draft, although we think it is a good idea, until it's been written up in a separate draft and there has been an opportunity for people to understand it and give feedback.

- adding an option of having Bob include, in message 2, a signed g^b value, his name (in the clear), and his certificate. This is the JFK [Aie01] idea, and our motivation for considering it was to come up with a document that incorporated good ideas from everyone. If this would have resulted in a single document, we felt it was worth doing, but technically it seemed like adding this as an option would add more complexity than any functionality gained. As explained in the section on identity hiding, if a choice were made to do one mechanism or the other, we felt the IKEv2 mechanism that avoided a polling attack and hid both identities from passive attackers was preferable.

- Some people suggested that our choice of the name IKEv2 was unfortunate for two reasons. Some people thought it was "presumptious" to call it IKEv2 when there were other contenders. However, at the time, Dan Harkins was tasked to come out with a cleaned up and document-merged version of IKEv2. The other contenders appeared simultaneously with the IKEv2 draft. Other people thought the name was unfortunate because many people concluded, without even looking at the draft, that anything named IKEv2 would just be patching IKEv1, and everyone agreed that IKEv1 was terminally complex. The confusion was exacerbated by naming the document the same name (The Internet Key Exchange) as IKEv1, so many people couldn't even find "IKEv2" listed in the drafts. We considered changing the name, but given that the only cute name any of the authors could come up with was "DOI" (daughter of IKE), and given that IKEv2 by now had "brand recognition", we decided to stick with

the name IKEv2. However, we changed the title of the draft to "Proposal for the IKEv2 Protocol".

3.2 Changes we decided to make

- We added support for preshared keys. Many people were against preshared keys because they were so awkwardly supported in IKEv1. In particular, in main mode, the ID had to be the IP address in order for Bob to look up the relevant key and decrypt Alice's identity. And they assumed that having multiple authentication methods was one of the reasons IKEv1 became so complex. However, other people like the performance advantage and configuration ease of preshared keys. Given the structure of the IKEv2 exchange, it was trivial to support preshared keys, and in a way that allowed arbitrary identities. The only change is that the proof of identity, instead of being a signature on what you sent plus the other side's nonce, becomes an HMAC of the preshared key with what you sent plus the other side's nonce.

- We added the option of having Alice specify, in message 1, Bob's identity. This facilitates having multiple services co-located at an address. The idea had come up before, and was suggested in [Ker01]. This did require us to explicitly say that this field, if present, should be excluded from what Alice signs.

- We added explicit instructions for the syntax of integrity protection and encryption rather than pointing to the syntax in the ESP spec. Our laziness in the first draft of IKEv2 caused confusion because people assumed there was some dependency of IKEv2 on ESP. [Note from CWK: I would characterize this as an attempt to minimize redundant specification rather than laziness, but in any case it is now fixed].

- We changed what each side signs to be what they send in the first pair of messages (minus Bob's name, if Alice sends that), plus the other side's nonce.

- We added the ability to provide IDs in phase 2, so that IKE can set up process-to-process IPsec SAs.

- We believe IKEv2's switch from random message IDs to sequence numbers, and changing the cookies to recipient-specified SPIs fixes the reflection and replay attacks in IKEv1. However, just because it is "good cryptographic hygiene", we computed different cryptographic keys in the two directions of the IKE-SA.

- We added the nonces into the SKEYSEED_a, SKEYSEED_d, and SKEYSEED_e stew. Again this is "good cryptographic hygiene", and certainly is at

least as good as what was there. It might count as a "gratuitous change", and can be changed back to what it was with minimal effort. This addition was suggested by Tero Kivinen.

- We allow the port number of the other end to be something other than 500, so that the protocol would work through NAT.

- We made some clarifications. We noted that the INITIAL-CONTACT message should affect SAs only from the same IP address. We specified the order of the fields, as being required to always be in the same order. We specified that, after rekeying, the sequence numbers start again at 1.

- All previous proposals (IKEv1, IKEv2, JFK, and SIGMA) were vulnerable to an active attacker answering Alice's message with bogus information. Alice cannot distinguish a legitimate message 2 from a bogus message 2. In this draft we explain how Alice can protect herself from this attack, which is to be willing to continue negotiation with every reply she receives. Only the legitimate Bob will be able to send an acceptable message 4, so the multiple SA's-in-progress will only last until the SA is set up.

[Aie01] Aiello, W., Bellovin, S., Blaze, M., Canetti, R., Ioannidis, J., Keromytis, A., Reingold, O., "Just Fast Keying (JFK)", draft-ietf-ipsec-jfk-00.txt

[Bra96] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

[Har01] Harkins, D., Kaufman, C., and Perlman, R., "The Internet Key Exchange (IKE) Protocol, draft-ietf-ipsec-ikev2-00.txt

[Ker01] Keromytis, A., and Sommerfeld, B., "The 'suggested ID' extension for IKE, draft-keromytis-ike-id-00.txt.

[PK01] Perlman, R., and Kaufman, C., "Analysis of the IPsec key exchange Standard", WET-ICE Security Conference, MIT, 2001, http://sec.femto.org/wetice-2001/papers/radia-paper.pdf.

Authors' Addresses

Dan Harkins
dharkins@tibernian.com

Tibernian Systems

Charlie Kaufman
ckaufman@notesdev.ibm.com
IBM

Tero Kivinen
kivinen@ssh.fi
SSH Communications Security Crop
Fredrikinkatu 42
FIN-00100 Helsinki
Finland

Stephen Kent
kent@bbn.com
BBN Corporation

Radia Perlman
radia.perlman@sun.com
Sun Microsystems