Internet Engineering Task Force
IP Security Working Group
Internet Draft

Andrew Krywaniuk
Alcatel
June 30, 2002

Security Properties of the IPsec Protocol Suite
<draft-ietf-ipsec-properties-02.txt>

Status of this Memo

This document is a submission to the IETF Internet Protocol Security (IPsec) Working Group. Comments are solicited and should be addressed to the working group mailing list (ipsec@lists.tislabs.com) or to the editor.

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC 2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or made obsolete by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Copyright Notice

Abstract

This document describes the "security properties" of the IPsec architecture and protocols, including ESP, AH, and IKE.

By documenting these properties, we aim to provide a guide for users who wish to understand the abilities and limitations of the IPsec protocol suite. We also hope to provide motivation for future work in this area.

Table of Contents

1. Introduction

Before you can know where you are going, you must first know where you have been.

An analysis of IPsec by Counterpane researchers [Counterpane] complained that IPsec has a lack of clearly expressed design goals, and shows evidence of design by committee. We concur with these observations, in the sense that some features appear incomplete or are not used for the purpose for which they were intended. Part of the confusion comes from the fact that [ISAKMP] defines a large set of features; [IKE] only uses a subset of these features, but it does not clearly state which ones.

The IPsec working group has undertaken a project to redesign the IKE protocol in order to "simplify" it; there has also been talk of reducing the number of IPsec usage permutations by deprecating AH and/or tunnel mode. We believe that it is inappropriate to redesign a protocol until the existing protocol is well documented.

Perhaps IPsec is well understood by some, but frequent questions on the developers' mailing list confirm that one cannot become an IPsec expert merely by reading the RFCs. Much valuable information is buried deep in the list archives or in the minds of its designers.

Other protocol designers depend on IPsec for transport security; if they cannot clearly understand what security properties IPsec provides, they may use it incorrectly. The same could be said for IPsec users.

2. Specification of Requirements

This document shall use the keywords "MUST", "MUST NOT", "REQUIRED","SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED, "MAY", and "OPTIONAL" to describe requirements. They are to be interpreted as described in [Bradner].

3. General Approach

This document is not an introduction to IPsec, nor is it cryptography 101. It is merely a description of the security properties associated with one particular suite of security protocols.

Our intention is merely to document what exists today. In the few places where we discuss alternatives, they relate specifically to

known issues concerning the security properties of IPsec.

Some of this information in this memo is already available in the RFCs, some is not; this document collects it all into one place. Sometimes the RFCs are ambiguous, for example in the case where a feature is described in ISAKMP, but not used in IKE; here, we attempt to resolve that ambiguity.

The amount of space devoted to a particular property does not necessarily reflect on the importance of that property in the context of IPsec. For example, identity protection is discussed in some detail, even though its applicability is limited, precisely because the issues are complicated.

3.1 Terminology

For the purposes of this document, "the IPsec protocol suite" shall consist of RFCs 2401 through 2412, plus any other documents which we consider relevant. We assume the use of ESP and/or AH SAs, negotiated by IKE, and used according to the rules prescribed by [ARCH]. We do not cover specialized applications, such as multicast and alternate key exchange protocols.

The "security properties" we discuss include properties such as confidentiality, authentication, and resistance to Denial of Service (DoS). We only attempt to define properties that can be measured objectively. As such, we do not discuss such issues as technical merit, ease of use, or level of complexity. The document focuses more on IKE than on ESP/AH, since IKE appears to have more intricate security properties.

4. Confidentiality

Traffic confidentiality is one of the main reasons for using IPsec. For better or for worse, IPsec provides two completely independent implementations of encryption: one in IKE and one in ESP.

Obviously, in a network scenario not all data can be encrypted. Otherwise, it would be impossible to create SAs and route traffic. What data is encrypted and what data is not is a matter of some interest.

4.1 Encryption Coverage

ESP encryption covers protocol layers 4 and above (and, potentially, some of the so-called layer 3.5 protocols). The IP header and any additional lower-level headers are sent in the clear. If tunnel mode is used, the data in the inner header can be concealed, but some of that information will be copied into the outer header anyway, since it is needed for routing.

In IKE, a large portion of the data must be sent in the clear, simply to bootstrap the negotiation. For example, an attacker can see which transforms are being used in IKE. (Modern cryptological thinking postulates that revealing this kind of data is not a security weakness.) Once the key exchange is complete, subsequent IKE data is encrypted.

4.2 Traffic Flow

ESP hides such information as the layer 4 port and protocol, however some information about the traffic flow is leaked due to packet sizes. ESP allows an implementation to add padding to packets in order to conceal packet lengths; this is constrained to a maximum of 255 bytes. A future version of ESP may allow extra padding, and even completely bogus packets.

In tunnel mode, when an edge device is applying the encryption, a snooper is generally unable to determine which end nodes the router is proxying for. This situation is improved if a single tunnel mode SA is used to carry all traffic between two protected networks, rather than using separate SAs for each traffic flow.

Sending multiple traffic flows on a single SA allows a privileged attacker who is behind one of the IPsec gateways to launch an adaptive chosen plaintext attack against the encryption key, thus compromising all traffic sent between the two networks. However, modern ciphers are assumed to be resistant to this type of cryptanalysis. The IPsec RFCs (e.g. [DES]) suggest that you may reuse a block of ciphertext from another packet as an IV, but it is better to use a completely random IV for each packet because of the attack described in [Fluhrer].

4.3 Cryptographic Mumbo-Jumbo

Encryption in ESP and ISAKMP is typically deployed using the CBC (Cipher Block Chaining) mode of operation. CBC uses the previous block as an IV to the next block, which ensures that the IV is always pseudorandom. A random IV makes it next to impossible that two blocks of ciphertext will be the same.

CBC does not have the infinite error propagation property, which means that it does not protect against known-plaintext analysis. This is not to say that IPsec is vulnerable to known-plaintext attacks; all it means is that the chosen cipher must itself be secure against these attacks (as all modern ciphers are). Other modes of operation could be used and they would have different security properties.

In general, encryption should not be used as a substitute for authentication, although some new modes propose to combine the two. With block ciphers, an attacker is generally prevented from making a predictable change to the plaintext. This is not necessarily true for other types of ciphers, such as stream ciphers.

4.4 Identity Protection

IKE main mode purports to deliver a feature called identity protection, which means that the identities are not sent in the clear. This it does, but with some caveats. In order to complete the authentication, one side must reveal its identity first. In main mode with public key signatures, the initiator reveals his identity first; therefore, an active attacker who impersonates the responder can determine the initiatorÆs identity.

Even when the identity is protected, a host may need to send a "certificate request" in order to force the sender to include a certificate in a later message. The certificate request payload typically contains the name of the CA to be used, which reveals some limited information about the sender's identity to a passive snooper.

This limited form of identity protection can only be used with public key signature authentication. Due to the particular construction of SKEYID_e in the case of preshared keys, the identity must be sent in the clear in order to generate the encryption key. The drawback is that mobile clients using preshared keys donÆt get identity protection. We recommend that this be fixed in a future version of IKE.

In the case of public key encryption, both identities are protected, but protection for the responder is weak for two reasons: Firstly, in order for the initiator to know where to send the packet, the responder's identity must be linked to an IP address (this is true of all the authentication methods). Secondly, in the case where the initiator sends a hash of the identity, this hash is an identity-equivalent, in the sense that it uniquely correlates to an identity. This means that a snooper can correlate multiple SAs negotiated with the same identity because the hash will be the same. Also, if the snooper can guess which possible identities might correspond to the

responder, he can test his assumption because the hash does not contain any secret material.

In many cases, the IP address of the host implicitly describes the identity (e.g. the identity can be found by a DNS lookup); in these cases, identity protection is moot. Since the initiatorÆs identity is less likely to be implicit from an IP address than the responderÆs is, itÆs a shame that signature-based authentication provides higher protection to the responderÆs id. On the other hand, it is much easier to mount an attack against the responderÆs id than against the initiatorÆs.

In the case where the identity is sent in the clear, it could be a random binary string; IKE allows the transmission of unformatted identities using the ID_KEY_ID type. However, it would be desirable that the obfuscated identity not be an identity-equivalent, so that multiple logins by the same user could not be correlated. IKE does not provide this feature.

IKE also provides a feature called Identity PFS, in which every quick mode exchange uses a new phase 1 SA. [IKE] doesnÆt specify why one might want to do this, but it does theoretically allow the host to delete the identity of the peer from memory, thus ensuring that it is not revealed even if the physical security of the box is compromised. (Although it may be difficult to apply policy rules if the identity of the peer is not remembered.)

5. Authentication

Traffic encryption is not much use if the user at the other end is unknown or if the data could be forged. IPsec provides several types of authentication: packet authentication, exchange authentication, and host authentication.

5.1 Authentication Coverage

Each AH packet contains an HMAC; likewise for ESP, assuming that ESP authentication is being used. ESP authentication covers the entire payload of the IP packet. AH also covers the non-mutable fields in the header.

When tunnel mode is being used, AH has the same effective coverage as ESP, because the outer header is merely a transient routing header. If AH is being used to ensure that the header of the IP packet remains uncorrupted during transit, this is really only useful if any

of the intermediate routers which interpret the header are also privy to the AH key.

The IKE phase 1 hash covers sufficient material to bind the identity of the peer to unforgeable session data, such as the DH secret. However, phase 1 does not have full authentication coverage (a shortcoming which should be fixed in a future version of the protocol). Consequently, optional payloads, such as notify messages and vendor ids, are not authenticated by the exchange. Even if these payloads are part of an encrypted message, an attacker can still corrupt them without being detected.

Phase 2 messages are fully authenticated by an HMAC, with the exception of the ISAKMP header. Modifying the commit bit (in the header) is a potential DoS vulnerability.

5.2 Cryptographic Mumbo-Jumbo

The HMAC that is used for packet authentication is truncated. This limits the amount of information an attacker can gather by analyzing the output.

The HMAC functions that are used in IKE are also used as PRFs. Therefore, the output of the HMAC should always appear statistically random. Uri Blumenthal has stated that HMAC has never been proven to be an adequate PRF, although we have no specific reason to believe it isn't. [IKE] allows alternate PRFs to be used, but IANA has yet to assign any.

5.3 Host Authentication

Depending on the chosen authentication method, the host is authenticated in IKE phase 1 by the generation of a public key signature, an HMAC signature, or by a proof of possession of a decryption key. The obvious man in the middle attack is thwarted by including the Diffie-Hellman public keys in the HASH_I/HASH_R values which are generated, signed, or encrypted.

When authenticating with preshared keys, the strength of the authentication is based on the effective entropy of the secret. When authenticating with public key encryption, the strength of the authentication is based on the length of the public key. Likewise for public key signatures, but as an additional wrinkle the strength of the MAC algorithm is also important. Since all the inputs to the MAC are sent on the wire except possibly the IDs (which can be guessed), the strength of the PK signature is limited by the difficulty of

finding a collision in the MAC function.

5.4 User Authentication

While IKE provides a number of ways to identity the peer, there is no standardized interface to communicate this identity up to the application layer. Ultimately, all traffic-level authorization must be applied to IPs and ports. If an application doesn't have a mechanism to extract the authenticated id from the IKE process then the application layer will have to perform its own, separate authentication stage.

The upside of this limitation is that a simple username/password protocol is obviously more secure when it is sent across an ESP tunnel than when it is performed in the clear, especially since the use of IKE authentication rules out the possibility of a man-in-the-middle attack.

6. Key Generation

Key generation is the most important function of the quick mode exchange. Key generation is also part of phase 1, since ISAKMP needs its own session keys.

The properties of key generation are more complicated and harder to explain than most other security properties. Nonetheless, we will make an attempt.

6.1 Rekeying

One purpose of rekeying is to thwart cryptanalysis by limiting the amount of ciphertext that an attacker can examine, but the main purpose is simply to limit the consequences of a compromised key.

[IKE] defines two different types of lifetimes: time-based and traffic-based. Time-based lifetimes protect against the possibility that a key will be compromised by brute force; traffic-based lifetimes guard against attacks based on gathering ciphertext. Both lifetimes limit the amount of data that is vulnerable if a key is compromised.

[IKE] also proposes a third lifetime for phase 1 SAs, based on the number of quick modes used with this SA. The justification given for this lifetime is suspect because a PRF can provide keying material for a large number of random keys, and these keys are not revealed to

an attacker for analysis. Nonetheless, this lifetime makes sense because it correlates strongly with the volume of IPsec traffic.

When using strong ciphers with small block sizes (e.g. 3DES), the use of rekeying to thwart cryptanalysis becomes more important. Due to the birthday paradox, an attacker has a statistically significant chance of detecting a collision in the output stream after he collects about 2^(block size/2) blocks of encrypted data. If each block is 64 bits, this works out to 32 gigabytes of encrypted traffic.

6.2 Independence of Keying Material

The keys negotiated by IKE are derived from the Diffie-Hellman secret, some random session data, and possibly a preshared key. This information is run through a pseudo-random function in order to generate a key.

The keys generated by IKE are not derived directly from each other, nor are they reused for multiple purposes. Each encryption or authentication key is created by an HMAC-based PRF, which is keyed by a shared primitive key that is never sent on the wire.

The PRF used in IKE must be a strong one-way function. This means that even if one key is compromised, other keys created from the same DH secret cannot be cracked unless the PRF is reversed.

In all cases, entropy for the key derivation is added explicitly by means of a random nonce. The size of the nonce is not all that important, but it should be larger than the square of the number of keys that will be derived from the raw key material. (It does no good to make the nonce larger than the HMAC output.)

6.3 Cryptographic Mumbo-Jumbo

All the components of the key material, including the DH secret are HMAC'ed before they are used. This ensures that any analytical attack on the key exchange function will not directly translate into an analytical attack on the key generation function.

Wherever possible, the SKEYID is derived from a secret value other than g^xy (this is not possible in the case of public key signatures). In the case where keys are generated from each other (e.g. SKEYID_d -> SKEYID_a -> SKEYID_e), g^xy is reintroduced at every stage so that the key is always directly based on a shared primitive. One exception to this rule is noted below in section 6.7.

A more detailed description of the motivation for SKEYID construction is given in [Krawczyk]. Whether or not this elaborate derivation was entirely necessary is a contentious issue.

6.4 Perfect Forward Secrecy

The description of PFS in IKE is complicated because there are actually two different types of forward secrecy. PFS of the first kind means that compromise of the long-term credential (e.g. an RSA key) will not reveal any session keys. PFS of the second kind means that an active attack on the system (e.g. the box is hacked) will not reveal all of the expired session keys.

The original requirement for the IPsec WG was PFS of the first kind (an earlier protocol called SKIP did not have this property). IKE phase 1 automatically provides this feature because the key is based on a per-session DH secret. This is the most important type of forward secrecy, and it is not optional in IKE. Therefore, in the context of IKE, the term PFS usually refers to the second type of forward secrecy.

PFS of the second kind can be used to reduce the window of vulnerability even further. In the case of a break-in, it is desirable that only a limited amount of data should be compromised; we call this the forward secrecy window. If the forward secrecy window is 1 hour, then no session key should ever be kept in memory for more than 1 hour after it is first used (and this includes the key material from which this session key was derived).

Unfortunately, [IKE] does not explain the intended usage of the PFS feature. I suspect that the intention was that it should be off for the first quick mode exchange(s) and on for subsequent exchanges (e.g. rekeys). Why? Because you may have deleted SKEYID_D from memory in order to reduce the consequences of a break-in. In practice, most people implement PFS as an on/off setting, and the user must configure which setting is to be used for each connection.

Note that PFS does not significantly increase the security of IKE against long-term cryptanalysis. An attacker who can crack the phase 1 DH exchange can presumably crack a second DH exchange with equivalent work. And deriving each session key from a separate shared primitive is overkill because independence of keying material is already guaranteed by using a strong PRF. If you want to increase your resistance to cryptanalysis, a better solution would be to lengthen the modulus of the phase 1 DH group and the block size of the PRF.

What PFS does provide is a faster alternative to phase 1 rekeying. Repetition of the authentication stage may detect if a user's certificate is revoked, but this could be accomplished by other means (e.g. periodic CRL retrieval). The only case where this would not be possible is identity PFS, where the host deletes the peer's identity after the SAs are created. IKE also specifies a method for bulk negotiation of keys. This can be used to accomplish PFS without further DH negotiations because it allows the peers to rekey even after SKEID_D has been deleted. (In practice, this feature is not widely implemented.)

Whether or not phase 1 rekeying actually provides any additional security over PFS depends on how much information an attacker can gather if the box is physically compromised or otherwise hacked into. If the box is entirely compromised and the attacker can learn the host's RSA private key (or preshared key table), then all is lost and the host will be insecure until the private key is replaced. If the compromise is less atomic, and the attacker merely discovers SKEYID (or, more precisely, both SKEYID_E and SKEYID_A), then he can act as a man in the middle during subsequent quick mode negotiations.

6.5 Weak Keys

IKE mandates the use of weak key checks. In practice, this does not provide a significant benefit because weak keys are very unlikely to be generated randomly (and an attacker wonÆt be able to detect them if they are used).

6.6 Ad Hoc Groups

IKE allows the negotiation of ad hoc groups, either during phase 1, or after phase 1 using new group mode. To use new group mode, an implementation would have to trust the phase 1 group enough to use it in the short term, but not trust it for long term security.

According to [IKE], implementations are meant to verify the primality of a proposed group before using it. The implications of this statement are interesting. Presumably, this is to detect implementation errors in the peer, rather than malfeasance. Otherwise, it would also be pertinent to send an attribute describing the algorithm by which the group was chosen (e.g. a seed, which is hashed, and then used as the starting point in a search for a prime).

New Group mode is not often used in practice, but it provides a theoretical extra level of security. If everyone uses the same group, then an attacker can build a large database of known keys for that

group, thus amortizing the cost of a brute force search over many keys. In practice this attack is not as devastating as it sounds, since Diffie-Hellman keys are already large enough to defend against birthday attacks, such as the Pollard-Rho method.

6.7 Key Strength

All keys in IKE are derived from a PRF output. A PRF provides a theoretically completely random key. Assuming that the cipher algorithm is strong against analysis, the most significant attack is brute force. Therefore, the strength of a key is approximately proportional to the key length.

But the length of a key is not the only factor in determining its strength. The length of the encryption key must be large enough to thwart a direct attack, but the length of the DH secret used to generate the key is also important, as described in [Orman].

[Beaulieu] notes that a third factor comes into play if one attempts to derive a large encryption key from a small hash output. As described in [IKE], the key material for an ISAKMP SA may be "stretched" using the following algorithm:

```
Ka = K1 | K2 | K3
```

and

```
K1 = prf(SKEYID_e, 0)
K2 = prf(SKEYID_e, K1)
K3 = prf(SKEYID_e, K2)
```

This definition does not fully utilize the entropy of the DH secret and further constrains the strength of the key to the length of the HMAC output. A similar limitation applies to the keys generated by quick mode when PFS is not being used.

7. Denial of Service

IPsec provides some protection against denial of service attacks but also creates some new holes.

7.1 ESP Packet Spoofing

IPsec ESP/AH authentication provides strong protection against DoS because any spoofed packets will be identified and discarded. The

time lost to DoS is limited to the length of time required to verify an HMAC. ESP without authentication has less DoS protection because encryption is generally slower than authentication; also, with the CBC mode of operation, it is quite easy to form a corrupt packet which will pass ESP processing.

IKE does not dictate how SPI values should be chosen, but many implementations choose SPIs randomly. The fact that SPIs are random, and therefore unknown to an unprivileged attacker, provides additional protection against spoofing. If an authenticated user sends encrypted packets which cause DoS, the source of the attack will be obvious.

However, packets that are sent in the clear can still cause DoS. Obviously, some packets, most notably the first few packets of IKE, must still be sent in the clear.

7.2 Memory Consumption

ISAKMP reuses the stateless cookie idea from Photuris, but IKE does not provide a mode in which they can be used. (Anti-clogging cookies are meant to prevent state clogging attacks akin to the TCP SYN attack.)

There are multiple ways to extend IKE to allow stateless operation. One method is to add an optional cookie exchange when a DoS attack is detected. Another technique is to repeat the information from message 1 in message 3 of the exchange. [Huttunen] describes a proposal for optimizing this technique with the use of an encrypted "state payload."

7.3 Time Consumption

The key exchange and public key authentication operations of IKE phase 1 provide the greatest vehicle for DoS. An attacker can generate a fake key exchange or signature payload, forcing the responder to perform a time-consuming modular exponentiation operation (without significant work by the attacker).

IKE provides some protection against this attack in main mode by requiring an initial cookie exchange. Even though the cookie cannot be used for stateless operation, it performs a similar function as a nonce, proving the liveness of the initiator.

Aggressive Mode is vulnerable because the signature must be generated before the cookie exchange is complete. The DH exponentiation can be

delayed until the third packet is received, but at the cost of latency. Quick mode with PFS has a similar vulnerability if the exchange is replayed; again, the attack can be avoided (at the cost of latency) by delaying the computation.

These comments mainly apply to cases where authentication is tightly bound to authorization. In cases (e.g. a publicly accessible server on the Internet) where authorization is not important and the authentication stage is performed merely to ensure the confidentiality of the negotiated key, the user is essentially unauthenticated and he is free to launch any of a myriad of DoS attacks which we won't describe here.

IKE does not provide explicit protection against DDoS zombies. Countermeasures such as client puzzles exist, but there is no mechanism for using them with IKE.

7.4 Synchronization

The lack of full authentication coverage in some IKE messages can allow an active attacker to exploit synchronization issues. For example, he can set the commit bit in the ISAKMP header, causing one side to wait for a CONNECTED notification that may never come.

Alternately, he could add a vendor id to an IKE phase 1 message that would cause one side to enable a non-standard behaviour. Since the vendor id is not authenticated, this could cause one host to behave in a non-interoperable manner.

An attacker can potentially prevent the delivery of delete notifications or forge invalid SPI/cookie messages, which could cause one side to delete an SA or to believe that an SA has been deleted by the peer. By forcing a connection to be repeatedly torn down, the attacker can cause a host to waste CPU in frequent renegotiations, to deny service to a legitimate user, or to waste memory maintaining an SA after the peer has disconnected.

8. Miscellaneous

There are a few additional security properties that do not fall into the above categories.

8.1 Replay

Replay of ESP data is a vulnerability that depends on the upper layer protocol. Many session-based protocols will reject replayed data. ESP and AH packets contain an anti-replay counter which may optionally be checked. This counter is not time-based, so it does not prevent an attacker from intercepting all packets, storing them, and then selectively delivering them at a future time. Replay protection can only be used in conjunction with packet authentication.

ISAKMP does not provide explicit replay protection. Replay protection is accomplished in some exchanges (main mode, aggressive mode, quick mode) by sending a random value (e.g. a nonce) to the peer and having them return that value in a subsequent message. This technique is not possible with 1 or 2 message exchanges (new group mode, info mode). Also, replaying a quick mode exchange with PFS can cause DoS, as noted in section 7.3.

It has been suggested that an implementation needs to remember all message ids it receives in order to avoid replayed messages; we believe that a better general-purpose solution is to add a replay counter to ISAKMP packets. A logical way to do this would be to simply convert the random message id into a counter (the randomness of the message id is only needed for collision-resistance, and not for any essential security feature).

8.2 Repudiation

Authentication using either pre-shared keys or public key encryption has the repudiation property. Either side is capable of forging the entire exchange; therefore there is no reliable way to prove that the transaction took place.

Authentication using public key signatures does not provide full repudiation, but it doesnÆt provide explicit non-repudiation either. When Bob generates a signature, it proves that he talked to somebody, but not necessarily Alice. It is possible for Alice to encode a signed hash of her identity into a payload that will be signed by Bob during the course of the exchange. This would prove that Bob talked to Alice (or someone colluding with Alice), although not necessarily on purpose. Note that this does not prove to a third party that any data sent with the negotiated keys is genuine.

So for all intents and purposes, IKE provides repudiation of the phase 1 exchange, no matter which mode of authentication you use.

9. IANA Considerations

This document does not require any assigned numbers.

10. Security Considerations

The focus of this document is security; hence security considerations permeate this specification.

11. Notes

The authors would like to thank Radia Perlman, Olivier Paradiens, and Angelos Kerymitis for their comments which were incorporated into this draft.

12. References

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

[Beaulieu] Beaulieu, S., "Generating 3DES keys from SKEYID_e", IPsec mailing list, http://www.vpnc.org/ietf-ipsec/00.ipsec/msg01288.html, July 2001.

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

[Counterpane] Ferguson, Niels, and Schneier, Bruce, "A Cryptographic Evaluation of IPSec", http://www.counterpane.com, April 1999.

[DES] Madson, C., and N. Doraswamy, "The ESP DES-CBC Cipher Algorithm With Explicit IV", RFC 2405, November 1998

[Fluhrer]Fluhrer, S., "Suggested modification to AES privacy draft", IPsec mailing list, http://www.vpnc.org/ietf-ipsec/mail-archive/msg02598.html, January 2002.

[Huttunen] Huttunen, A., "Re: Future ISAKMP Denial of Service Vulnerablity Needs Addressing", IPsec mailing list, http://www.vpnc.org/ietf-ipsec/00.ipsec/msg00160.html, January 2000.

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

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

[Krawczyk] Krawczyk, H., "Rationale for the definitions of SKEYID", IPsec mailing list, http://www.vpnc.org/ietf-ipsec/mail-archive/msg00844.html, June 2001.

[Orman] Orman, H., Hoffman, P., "Determining Strengths for Public Keys Used For Exchanging Symmetric Keys", draft-orman-public-key-lengths-02.txt, March 19, 2001 (work in progress)

Authors' Addresses

Andrew Krywaniuk
Alcatel Networks Corporation
600 March Road
Kanata, ON
Canada, K2K 2E6
+1 (613) 784-4237
E-mail: andrew.krywaniuk@alcatel.com

The IPsec working group can be contacted via the IPsec working group's mailing list (ipsec@lists.tislabs.com) or through its chairs:

Theodore Y. Ts'o
tytso@MIT.EDU
Massachusetts Institute of Technology

Barbara Fraser
byfraser@cisco.com
Cisco Systems

Expiration

This document expires January 30th, 2003.