Internet Draft
June 2002
Expiration Date: December 2002

IPsec Working Group
S. Frankel, NIST
S. Kelly, Black Storm Networks
R. Glenn, NIST

The AES Cipher Algorithm and Its Use With IPsec

<draft-ietf-ipsec-ciph-aes-cbc-04.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 editors.

Distribution of this memo is unlimited.

Abstract

This document describes the use of the AES Cipher Algorithm in Cipher Block Chaining Mode, with an explicit IV, as a confidentiality mechanism within the context of the IPsec Encapsulating Security Payload (ESP).

Table of Contents

1. Introduction

As the culmination of a four-year competitive process, NIST (the National Institute of Standards and Technology) has selected the AES (Advanced Encryption Standard), the successor to the venerable DES. The competition was an open one, with public participation and comment solicited at each step of the process. The AES [AES], formerly known as Rijndael, was chosen from a field of five finalists.

The AES selection was made on the basis of several characteristics:

+ security
+ unclassified
+ publicly disclosed
+ available royalty-free worldwide
+ capable of handling a block size of at least 128 bits
+ at a minimum, capable of handling key sizes of 128, 192, and 256 bits
+ computational efficiency and memory requirements on a variety of software and hardware, including smart cards
+ flexibility, simplicity and ease of implementation

The AES will be the government's designated encryption cipher. The expectation is that the AES will suffice to protect sensitive (unclassified) government information at least until the next century. It is also expected to be widely adopted by businesses and financial institutions.

It is the intention of the IETF IPsec Working Group that AES will eventually be adopted as the default IPsec ESP cipher and will obtain the status of MUST be included in compliant IPsec implementations.

The remainder of this document specifies the use of the AES within the context of IPsec ESP. For further information on how the various pieces of ESP fit together to provide security services, refer to [ARCH], [ESP], and [ROAD].

1.1 Specification of Requirements

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",

"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" that appear in this document are to be interpreted as described in [RFC-2119].

2. The AES Cipher Algorithm

All symmetric block cipher algorithms share common characteristics and variables, including mode, key size, weak keys, block size, and rounds. The following sections contain descriptions of the relevant characteristics of the AES cipher.

2.1 Mode

NIST has defined 5 modes of operation for AES and other FIPS-approved ciphers [MODES]: CBC (Cipher Block Chaining), ECB (Electronic Code-Book), CFB (Cipher FeedBack), OFB (Output FeedBack) and CTR (Counter). The CBC mode is well-defined and well-understood for symmetric ciphers, and is currently required for all other ESP ciphers.

This document specifies the use of the AES cipher in CBC mode within ESP. This mode requires an Initialization Vector (IV) that is the same size as the block size. Use of a randomly generated IV prevents generation of identical ciphertext from packets which have identical data that spans the first block of the cipher algorithm's block size.

The IV is XOR'd with the first plaintext block before it is encrypted. Then for successive blocks, the previous ciphertext block is XOR'd with the current plaintext, before it is encrypted.

More information on CBC mode can be obtained in [MODES, CRYPTO-S]. For the use of CBC mode in ESP with 64-bit ciphers, see [CBC].

2.2 Key Size

Some cipher algorithms allow for variable sized keys, while others only allow specific, pre-defined key sizes. The length of the key typically correlates with the strength of the algorithm; thus larger keys are usually harder to break than shorter ones.

This document specifies the default (i.e. MUST be supported) key size for the AES cipher algorithm. The default key size that implementations MUST support for IPsec is 128 bits. In addition, implementations MAY support key sizes of 192 and 256 bits.

2.3 Weak Keys

At the time of writing this document there are no known weak keys for the AES.

Some cipher algorithms have weak keys or keys that MUST not be used due to their interaction with some aspect of the cipher's definition. If weak keys are discovered for the AES, then weak keys SHOULD be checked for and discarded when using manual key management. When using dynamic key management, such as [IKE], weak key checks SHOULD NOT be performed as they are seen as an unnecessary added code complexity that could weaken the intended security [EVALUATION].

2.4 Block Size and Padding

The AES uses a block size of sixteen octets (128 bits).

Padding is required by the AES to maintain a 16-octet (128-bit) blocksize. Padding MUST be added, as specified in [ESP], such that the data to be encrypted (which includes the ESP Pad Length and Next Header fields) has a length that is a multiple of 16 octets.

Because of the algorithm specific padding requirement, no additional padding is required to ensure that the ciphertext terminates on a 4-octet boundary (i.e. maintaining a 16-octet blocksize guarantees that the ESP Pad Length and Next Header fields will be right aligned within a 4-octet word). Additional padding MAY be included, as specifed in [ESP], as long as the 16-octet blocksize is maintained.

2.5 Rounds

This variable determines how many times a block is encrypted. While this variable MAY be negotiated, a default value MUST always exist when it is not negotiated. Within IPsec, the AES MUST support 10 rounds, corresponding to the mandatory 128-bit keysize.

The AES's default number of rounds is 12 for a 192-bit keysize and 14 for a 256-bit keysize.

2.6 Additional Information

AES was invented by Joan Daemen from Banksys/PWI and Vincent Rijmen from ESAT-COSIC, both in Belgium, and is available world-wide on a royalty-free basis. It is not covered by any patents, and the Rijndael homepage contains the following statement: "Rijndael is available for free. You can use it for whatever purposes you want, irrespective of whether it is accepted as AES or not." AES's description can be found in [AES]. The Rijndael homepage is: http://www.esat.kuleuven.ac.be/~rijmen/rijndael/.

The AES homepage, http://www.nist.gov/aes, contains a wealth of information about the AES, including a definitive description of the AES algorithm, performance statistics, test vectors and intellectual property information. This site also contains information on how to obtain an AES reference implementation from NIST.

2.7 Performance

For a comparison table of the estimated speeds of AES and other cipher algorithms, please see [PERF-1], [PERF-2], [PERF-3], or [PERF-4]. The AES homepage has pointers to other analyses.

3. ESP Payload

The ESP payload is made up of the IV followed by raw cipher-text. Thus the payload field, as defined in [ESP], is broken down according to the following diagram:

```
+---------------+---------------+---------------+---------------+
|                                                               |
+               Initialization Vector (16 octets)               +
|                                                               |
+---------------+---------------+---------------+---------------+
|                                                               |
~ Encrypted Payload (variable length, a multiple of 16 octets)  ~
|                                                               |
+---------------------------------------------------------------+
```

The IV field MUST be the same size as the block size of the cipher algorithm being used. The IV MUST be chosen at random, and MUST be unpredictable.

Including the IV in each datagram ensures that decryption of each received datagram can be performed, even when some datagrams are dropped, or datagrams are re-ordered in transit.

To avoid CBC encryption of very similar plaintext blocks in different packets, implementations MUST NOT use a counter or other low-Hamming distance source for IVs.

3.1 ESP Algorithmic Interactions

Currently, there are no known issues regarding interactions between the AES and other aspects of ESP, such as use of certain authentication schemes.

3.2 Keying Material

The minimum number of bits sent from the key exchange protocol to the ESP algorithm must be greater than or equal to the key size.

The cipher's encryption and decryption key is taken from the first <x> bits of the keying material, where <x> represents the required key size.

4. Test Vectors

The first 4 test cases test AES-CBC encryption. Each test case includes the key, the plaintext, and the resulting ciphertext. The values of keys and data are either hexadecimal numbers (prefixed by "0x") or ASCII character strings (surrounded by double quotes). If a value is an ASCII character string, then the AES-CBC computation for the corresponding test case DOES NOT include the trailing null character ('\0') of the string. The computed cyphertext values are all hexadecimal numbers.

The last 4 test cases illustrate sample ESP packets using AES-CBC for encryption. All data are hexadecimal numbers (not prefixed by "0x").

These test cases were verified using 2 independent implementations: the NIST AES-CBC reference implementation and an implementation provided by the authors of the Rijndael algorithm

```
(http://csrc.nist.gov/encryption/aes/rijndael/
                      rijndael-unix-refc.tar).

Case #1: Encrypting 16 bytes (1 blocks) using AES-CBC with 128-bit key
Key       : 0x06a9214036b8a15b512e03d534120006
IV        : 0x3dafba429d9eb430b422da802c9fac41
Plaintext : "Single block msg"
Ciphertext: 0xe353779c1079aeb82708942dbe77181a

Case #2: Encrypting 32 bytes (2 blocks) using AES-CBC with 128-bit key
Key       : 0xc286696d887c9aa0611bbb3e2025a45a
IV        : 0x562e17996d093d28ddb3ba695a2e6f58
Plaintext : 0x000102030405060708090a0b0c0d0e0f
              101112131415161718191a1b1c1d1e1f
Ciphertext: 0xd296cd94c2cccf8a3a863028b5e1dc0a
              7586602d253cfff91b8266bea6d61ab1

Case #3: Encrypting 48 bytes (3 blocks) using AES-CBC with 128-bit key
Key       : 0x6c3ea0477630ce21a2ce334aa746c2cd
IV        : 0xc782dc4c098c66cbd9cd27d825682c81
Plaintext : "This is a 48-byte message (exactly 3 AES blocks)"
Ciphertext: 0xd0a02b3836451753d493665d33f0e886
              2dea54cdb293abc7506939276772f8d5
              021c19216bad525c8579695d83ba2684

Case #4: Encrypting 64 bytes (4 blocks) using AES-CBC with 128-bit key
Key       : 0x56e47a38c5598974bc46903dba290349
IV        : 0x8ce82eefbea0da3c44699ed7db51b7d9
Plaintext : 0xa0a1a2a3a4a5a6a7a8a9aaabacadaeaf
              b0b1b2b3b4b5b6b7b8b9babbbcbdbebf
              c0c1c2c3c4c5c6c7c8c9cacbcccdcecf
              d0d1d2d3d4d5d6d7d8d9dadbdcdddedf
Ciphertext: 0xc30e32ffedc0774e6aff6af0869f71aa
              0f3af07a9a31a9c684db207eb0ef8e4e
              35907aa632c3ffdf868bb7b29d3d46ad
              83ce9f9a102ee99d49a53e87f4c3da55

Case #5: Sample transport-mode ESP packet (ping 192.168.123.100)
Key: 90d382b4 10eeba7a d938c46c ec1a82bf
SPI: 4321
Source address: 192.168.123.3
Destination address: 192.168.123.100
Sequence number: 1
IV: e96e8c08 ab465763 fd098d45 dd3ff893

Original packet:
IP header (20 bytes): 45000054 08f20000 4001f9fe c0a87b03 c0a87b64
Data (64 bytes):
```

```
08000ebd a70a0000 8e9c083d b95b0700 08090a0b 0c0d0e0f 10111213 14151617
18191a1b 1c1d1e1f 20212223 24252627 28292a2b 2c2d2e2f 30313233 34353637
```

Augment data with:
Padding: 01020304 05060708 090a0b0c 0d0e
Pad length: 0e

```
Next header: 01 (ICMP)

Pre-encryption Data with padding, pad length and next header (80 bytes):
08000ebd a70a0000 8e9c083d b95b0700 08090a0b 0c0d0e0f 10111213 14151617
18191a1b 1c1d1e1f 20212223 24252627 28292a2b 2c2d2e2f 30313233 34353637
01020304 05060708 090a0b0c 0d0e0e01

Post-encryption packet with SPI, Sequence number, IV:
IP header: 4500007c 08f20000 4032f9a5 c0a87b03 c0a87b64
SPI/Seq #: 00004321 00000001
IV: e96e8c08 ab465763 fd098d45 dd3ff893
Encrypted Data (80 bytes):
f663c25d 325c18c6 a9453e19 4e120849 a4870b66 cc6b9965 330013b4 898dc856
a4699e52 3a55db08 0b59ec3a 8e4b7e52 775b07d1 db34ed9c 538ab50c 551b874a
a269add0 47ad2d59 13ac19b7 cfbad4a6

Case #6: Sample transport-mode ESP packet (ping -p 77 -s 20 192.168.123.100)
Key: 90d382b4 10eeba7a d938c46c ec1a82bf
SPI: 4321
Source address: 192.168.123.3
Destination address: 192.168.123.100
Sequence number: 8
IV: 69d08df7 d203329d b093fc49 24e5bd80

Original packet:
IP header (20 bytes): 45000030 08fe0000 4001fa16 c0a87b03 c0a87b64
Data (28 bytes):
0800b5e8 a80a0500 a69c083d 0b660e00 77777777 77777777 77777777

Augment data with:
Padding: 0102
Pad length: 02
Next header: 01 (ICMP)

Pre-encryption Data with padding, pad length and next header (32 bytes):
0800b5e8 a80a0500 a69c083d 0b660e00 77777777 77777777 77777777 01020201

Post-encryption packet with SPI, Sequence number, IV:
IP header: 4500004c 08fe0000 4032f9c9 c0a87b03 c0a87b64
SPI/Seq #: 00004321 00000008
IV: 69d08df7 d203329d b093fc49 24e5bd80
Encrypted Data (32 bytes):
f5199588 1ec4e0c4 488987ce 742e8109 689bb379 d2d750c0 d915dca3 46a89f75

Case #7: Sample tunnel-mode ESP packet (ping 192.168.123.200)
Key: 01234567 89abcdef 01234567 89abcdef
SPI: 8765
Source address: 192.168.123.3
Destination address: 192.168.123.200
```

Sequence number: 2
IV: f4e76524 4f6407ad f13dc138 0f673f37

Original packet:
IP header (20 bytes): 45000054 09040000 4001f988 c0a87b03 c0a87bc8

```
Data (64 bytes):
08009f76 a90a0100 b49c083d 02a20400 08090a0b 0c0d0e0f 10111213 14151617
18191a1b 1c1d1e1f 20212223 24252627 28292a2b 2c2d2e2f 30313233 34353637

Augment data with:
Padding: 01020304 05060708 090a
Pad length: 0a
Next header: 04 (IP-in-IP)

Pre-encryption Data with original IP header, padding, pad length and
                         next header (96 bytes):
45000054 09040000 4001f988 c0a87b03 c0a87bc8 08009f76 a90a0100 b49c083d
02a20400 08090a0b 0c0d0e0f 10111213 14151617 18191a1b 1c1d1e1f 20212223
24252627 28292a2b 2c2d2e2f 30313233 34353637 01020304 05060708 090a0a04

Post-encryption packet with SPI, Sequence number, IV:
IP header: 4500008c 09050000 4032f91e c0a87b03 c0a87bc8
SPI/Seq #: 00008765 00000002
IV: f4e76524 4f6407ad f13dc138 0f673f37
Encrypted Data (96 bytes):
773b5241 a4c44922 5e4f3ce5 ed611b0c 237ca96c f74a9301 3c1b0ea1 a0cf70f8
e4ecaec7 8ac53aad 7a0f022b 859243c6 47752e94 a859352b 8a4d4d2d ecd136e5
c177f132 ad3fbfb2 201ac990 4c74ee0a 109e0ca1 e4dfe9d5 a100b842 f1c22f0d

Case #8: Sample tunnel-mode ESP packet (ping -p ff -s 40 192.168.123.200)
Key: 01234567 89abcdef 01234567 89abcdef
SPI: 8765
Source address: 192.168.123.3
Destination address: 192.168.123.200
Sequence number: 5
IV: 85d47224 b5f3dd5d 2101d4ea 8dffab22

Original packet:
IP header (20 bytes): 45000044 090c0000 4001f990 c0a87b03 c0a87bc8
Data (48 bytes):
0800d63c aa0a0200 c69c083d a3de0300 ffffffff ffffffff ffffffff ffffffff
ffffffff ffffffff ffffffff ffffffff

Augment data with:
Padding: 01020304 05060708 090a
Pad length: 0a
Next header: 04 (IP-in-IP)

Pre-encryption Data with original IP header, padding, pad length and
                         next header (80 bytes):
45000044 090c0000 4001f990 c0a87b03 c0a87bc8 0800d63c aa0a0200 c69c083d
a3de0300 ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
ffffffff 01020304 05060708 090a0a04
```

```
Post-encryption packet with SPI, Sequence number, IV:
IP header: 4500007c 090d0000 4032f926 c0a87b03 c0a87bc8
SPI/Seq #: 00008765 00000005
IV: 85d47224 b5f3dd5d 2101d4ea 8dffab22
Encrypted Data (80 bytes):
```

15b92683 819596a8 047232cc 00f7048f e45318e1 1f8a0f62 ede3c3fc 61203bb5
0f980a08 c9843fd3 a1b06d5c 07ff9639 b7eb7dfb 3512e5de 435e7207 ed971ef3
d2726d9b 5ef6affc 6d17a0de cbb13892

5. IKE Interactions

5.1 Phase 1 Identifier

For Phase 1 negotiations, IANA has assigned an Encryption Algorithm ID of 7 for AES-CBC.

5.2 Phase 2 Identifier

For Phase 2 negotiations, IANA has assigned an ESP Transform Identifier of 12 for ESP_AES.

5.3 Key Length Attribute

Since the AES allows variable key lengths, the Key Length attribute MUST be specified in both a Phase 1 exchange [IKE] and a Phase 2 exchange [DOI].

5.4 Diffie-Hellman Groups

The Diffie-Hellman algorithm is the basis of cryptographic key exchange within IPsec. The algorithm may be implemented using either "MODP" (modulus-exponent) groups or "EC" (elliptic curve) groups. The general procedure is as follows: the initiator chooses a random exponent x with K bits of entropy that is 2K bits in length (the K bits may be hashed to produce 2K bits), and then computes g^x using the group operation:

X = g^x

For MODP the group operation is modular multiplication, while for EC the operation is point addition on the curve. The notation "g^x" means "iterate the group operation x times". X is then sent to the responder. The responder chooses a secret number y, and similarly computes

Y = g^y

which is in turn sent to the initiator. At this point, both the initiator and responder may compute a shared secret value by combining their own secret value with the exponential and applying the group operation:

$$Z = g^{(xy)} = Y^x = X^y$$

From Z, both derive identical cryptographic keys.

This description is simplified in the interest of brevity, and an in-depth description of this mechanism is beyond the scope of this memo. For further details, refer to the wealth of published literature on this topic.

5.4.1 Relative Strength

The relative strength of the encryption keys derived via the Diffie-Hellman exchange may be characterized in terms the randomness of the participant's exponents and the strength of the Diffie-Hellman group; if an exponent has at least 128 completely random bits, it is said to have 128-bits of "entropy". If the Diffie-Hellman group cannot be broken in less time than searching a 128-bit key space, then the derived 128-bit key is said to have 128 bits of "strength". For an in-depth discussion regarding relative strength of values derived from DH exchanges, see [KEYLEN].

In some cases, one may choose to settle for an amount of entropy which is less than that of a completely random key of the given size. There are numerous reasons for making such a choice, among which might include a concern for the computational effort required to complete the key exchange. For example, the following table lists recommended modulus and exponent sizes for various key lengths using either MODP or EC groups.

Key Size	Exponent Size	Modulus Size	Group Type
128	256	3240	MODP
192	384	7945	MODP
256	512	15430	MODP
128	248	248	EC2N
192	376	376	EC2N
256	504	504	EC2N

NOTE: This table is based on Section 4.5 in [KEYLEN]

Note that the sizes of the moduli and exponents for the MODP groups in the table above are very large, and the computational effort required to complete the exponentiation and modulo operations with such large values is quite significant using hardware commonly available in the year 2002. If such considerations are deemed important, then keys larger than 128 bits SHOULD NOT be used. Further, if it is determined that less than 128 bits of strength will suffice for the security requirements of the given application, then smaller exponents and moduli may be used.

[GROUPS] defines four additional Diffie-Hellman MODP groups for IKE. Two of these groups, a 3072-bit MODP group and a 4096-bit MODP group, could be used to establish 128-bit AES keys. [IKE-ECC] defines four additional Diffie-Hellman ECC groups for IKE. Two of these groups, Group 8 and 9, both of which are 283-bit ECC groups, could be used to establish 128-bit AES keys. Additional information about the relationship between the group governing a Diffie-Hellman exchange and the symmetric keys derived from the exchange can be found in [KEYLEN].

5.5 Hash Algorithm Considerations

A companion competition, to select the successor to SHA-1, the widely-used hash algorithm, recently concluded. The resulting hashes, called SHA-256, SHA-384 and SHA-512 [SHA2-1, SHA2-2] are capable of producing output of three different lengths (256, 384 and 512 bits), sufficient for the generation (within IKE) and authentication (within ESP) of the three AES key sizes (128, 192 and 256 bits). IANA has already assigned Phase 1 Hash Algorithm values of 4, 5 and 6 to SHA2-256, SHA2-384, and SHA2-512. IANA has also assigned AH Transform Identifiers of 5, 6 and 7 to AH_SHA2-256, AH_SHA2-384, and AH_SHA2-512.)

However, HMAC-SHA-1 [HMAC-SHA] and HMAC-MD5 [HMAC-MD5] are currently considered of sufficient strength to serve both as IKE generators of 128-bit AES keys and as ESP authenticators for AES encryption using 128-bit keys.

6. Security Considerations

Implementations are encouraged to use the largest key sizes they can when taking into account performance considerations for their particular hardware and software configuration. Note that encryption necessarily impacts both sides of a secure channel, so such consideration must take into account not only the client side, but the server as well. However, a key size of 128 bits is considered secure for the foreseeable future.

For more information regarding the necessary use of random IV values, see [CRYPTO-B].

For further security considerations, the reader is encouraged to read [AES].

7. IANA Considerations

IANA has assigned Encryption Algorithm ID 7 to AES-CBC.
IANA has assigned ESP Transform Identifier 12 to ESP_AES.

8. Intellectual Property Rights Statement

Pursuant to the provisions of [RFC-2026], the authors represent that they have disclosed the existence of any proprietary or intellectual

9. Acknowledgments

Portions of this text, as well as its general structure, were unabashedly lifted from [CBC].

The authors want to thank Hilarie Orman for providing expert advice (and a sanity check) on key sizes, requirements for Diffie-Hellman groups, and IKE interactions. We also thank Scott Fluhrer for his helpful comments and recommendations.

10. References

[AES] NIST, FIPS PUB 197, "Advanced Encryption Standard (AES)," November 2001.
http://csrc.nist.gov/publications/fips/fips197/fips-197.{ps,pdf}

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

[CBC] Pereira, R. and R. Adams, "The ESP CBC-Mode Cipher Algorithms," RFC 2451, November 1998.

[CRYPTO-B] Bellovin, S., "Probable Plaintext Cryptanalysis of the IP Security Protocols", Proceedings of the Symposium on Network and Distributed System Security, San Diego, CA, pp. 155-160, February 1997.
http://www.research.att.com/~smb/papers/probtxt.{ps, pdf}

[CRYPTO-S] B. Schneier, "Applied Cryptography Second Edition", John Wiley & Sons, New York, NY, 1995, ISBN 0-471-12845-7.

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

[ESP] Kent, S. and R. Atkinson, "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998.

[EVALUATION] Ferguson, N. and B. Schneier, "A Cryptographic Evaluation of IPsec," Counterpane Internet Security, Inc., January 2000. http://www.counterpane.com/ipsec.{pdf,ps.zip}

[GROUPS] Kivinen, T. and M. Kojo, "More MODP Diffie-Hellman groups for IKE," draft-ietf-ipsec-ike-modp-groups-00.txt, October 2000.

[HMAC-MD5] Madson, C. and R. Glenn, "The Use of HMAC-MD5-96 within ESP and AH," RFC 2403, November 1998.

[HMAC-SHA] Madson, C. and R. Glenn, "The Use of HMAC-SHA-1-96 within ESP and AH," RFC 2404, November 1998.

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

[IKE-ECC] Panjwani, P. and Y. Poeluev, "Additional ECC Groups For IKE," draft-ietf-ipsec-ike-ecc-groups-02.txt, May 2000.

[KEYLEN] Orman, H. and P. Hoffman, "Determining Strengths For Public Keys Used For Exchanging Symmetric Keys," draft-orman-public-key-lengths-01.txt, August 2000.

[MODES] Dworkin, M., "Recommendation for Block Cipher Modes of Operation: Methods and Techniques," NIST Special Publication 800-38A, December 2001. http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf

[PERF-1] Bassham, L. III, "Efficiency Testing of ANSI C Implementations of Round1 Candidate Algorithms for the Advanced Encryption Standard." http://csrc.nist.gov/encryption/aes/round1/r1-ansic.pdf

[PERF-2] Lipmaa, Helger, "AES/Rijndael: speed." http://www.tcs.hut.fi/~helger/aes/rijndael.html

[PERF-3] Nechvetal, J., E. Barker, D. Dodson, M. Dworkin, J. Foti and E. Roback, "Status Report on the First Round of the Development of the Advanced Encryption Standard."

http://csrc.nist.gov/encryption/aes/round1/r1report.pdf

[PERF-4] Schneier, B., J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson, "Performance Comparison of the AES Submissions."

http://www.counterpane.com/aes-performance.{pdf,ps.zip}

[RFC-2026] Bradner, S., "The Internet Standards Process -- Revision 3", RFC2026, October 1996.

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

[ROAD] Thayer, R., N. Doraswamy and R. Glenn, "IP Security Document Roadmap", RFC 2411, November 1998.

[SHA2-1] NIST, Draft FIPS PUB 180-2 "Specifications for the Secure Hash Standard," May 2001. http://csrc.nist.gov/encryption/shs/dfips-180-2.pdf

[SHA2-2] "Descriptions of SHA-256, SHA-384, and SHA-512." http://csrc.nist.gov/cryptval/shs/sha256-384-512.pdf

11. Authors' Addresses

Sheila Frankel
NIST
820 West Diamond Ave.
Room 680
Gaithersburg, MD 20899
Phone: +1 (301) 975-3297
Email: sheila.frankel@nist.gov

Scott Kelly
Black Storm Networks
250 Cambridge Ave
Palo Alto CA 94304
Phone: +1 (650) 617-2934
Email: scott@bstormnetworks.com

Rob Glenn
NIST
820 West Diamond Ave.
Room 605
Gaithersburg, MD 20899
Phone: +1 (301) 975-3667
Email: rob.glenn@nist.gov

The IPsec working group can be contacted through the chairs:

Barbara Fraser
Cisco Systems Inc.

Email: byfraser@cisco.com

Theodore Ts'o
Massachusetts Institute of Technology
Email: tytso@mit.edu

12. Full Copyright Statement