Performance analysis of IPSec protocol: Encryption and authentication

Conference Paper · February 2002

DOI: 10.1109/ICC.2002.997033 · Source: IEEE Xplore

عمر الكيلاني
Tennessee Technological University
59 PUBLICATIONS **317** CITATIONS
SEE PROFILE

Mustafa M Matalgah
University of Mississippi
122 PUBLICATIONS **774** CITATIONS
SEE PROFILE

Ghulam Chaudhry
University of Missouri - Kansas City
78 PUBLICATIONS **238** CITATIONS
SEE PROFILE

Deep Medhi
University of Missouri - Kansas City
275 PUBLICATIONS **2,688** CITATIONS
SEE PROFILE

Performance Analysis of IPSec Protocol: Encryption and Authentication

O. Elkeelany*, M. M. Matalgah+, K. P. Sheikh+, M. Thaker+, G. Chaudhry*, D. Medhi* , J. Qaddour+

*University of Missouri- Kansas City
5100 Rockhill Road
Kansas City, MO 64110

+Sprint
6450 Sprint pkwy
Overland Park, KS 66251

Abstract- **IPSec provides two types of security algorithms, symmetric encryption algorithms (e.g. Data Encryption Standard DES) for encryption, and one-way hash functions (e.g., Message Digest MD5 and Secured Hash Algorithm SHA1) for authentication. This paper presents performance analysis and comparisons between these algorithms in terms of time complexity and space complexity. Parameters considered are processing power and input size. The analysis results revealed that HMAC-MD5 can be sufficient for the authentication purposes rather than using the more complicated HMAC-SHA1 algorithm. In encryption applications, authentication should be combined with DES.**

I. INTRODUCTION

IPSec is an IP security protocol that provides a solution for Secured Tunneling for ensuring Authenticated and Encrypted data flow. It Provides security at Layer 3 (IP Network Layer) by enabling a system to select required security protocols, and determine the algorithm(s) for encryption. IPSec has two security protocols, IP Authentication Header (IPSec AH) [1] and IP Encapsulating Security Payload (IPSec ESP) [2]. IPSec provides two types of security algorithms, symmetric encryption algorithms (e.g. Data Encryption Standard DES) [3][4] and One-way hash functions (e.g., Message Digest MD5 and Secured Hash Algorithm SHA-1) [5][6].

IPSec is used to secure tunnels against false data origins, and encrypt traffic against unwanted network passive or active intruders from listening or modifying actions. IPSec can be used between pairs of "Security Gateways," pairs of peer "hosts", or a combination (e.g. between a host and corporate equipment through security Gateways.)

IPSec tunnel mode can be used in the application of portable wireless virtual private networks as in [7]. Authentication is used for granting user access to his corporate network. Encryption is used to transfer user critical information through unreliable medium of the IP network.

The rest of the paper is organized as follows. IPSec overview is first presented. Then analytical analysis is provided for IPSec encryption and authentication algorithms, including space complexity, time complexity, and protocol limitations. Last section is summary and conclusion.

II. IPSEC OVERVIEW

IPSec has two modes of operation: Transport mode, and Tunnel mode. The transport mode provides upper layer protection (transport layer). It applies to pairs of peer hosts. Also in this mode, traffic goes to the ultimate destination directly. Whereas in the Tunnel mode, a tunneled IP protection is provided (via Gateways), in which traffic have to pass through a gateway to access ultimate destination.

In this section, we present integrity types used in IPSec and an overview of IPSec AH and IPSec ESP.

A. *Integrity*

There are three types of integrity in the context of IPSec:

1-Connectionless integrity, that detects modifications of a single IP packet by the use of an algorithm specific Integrity Check Value (ICV).
2-Anti-replay integrity, that detects duplicate IP packets at the receiver end by the use of sequence numbers.
3-Connection-oriented integrity, that detects lost or reordered IP packets at the receiver end by the use of sequence numbers too.

B. *IPSec Authentication Header (AH)*

IPSec AH Provides connectionless integrity, data origin authentication, and anti-replay integrity. The later is optional and not enforced at the receiver's end.

Figure 1 depicts the IP AH header format. The "Next Header" field is of 8-bit size and specifies the type of the transport protocol used in the upper layer. The "Payload Length" field is also an 8-bit size, and contains the IPSec header length in words (32bit) minus 2 words, e.g. 3+3-2= 4, if authentication data is 3 words (96bits). The sender always transmits the "Sequence Number" field (32 bits), but the receiver might optionally act on it. Finally, the "Authentication Data" field, variable size, multiple of 32 bits, ICV for the attached packet (Including the AH header itself). "Reserved" bits Must Be Zero (MBZ).

Next Header	Payload Lenghth	Reserved [12] (MBZ)
Security Parameter Index (SPI) [32]		
Sequence Number field [32]		
Authentication Data		

Figure 1. IPSec AH Header format

Figure 2 depicts the coverage of authentication protection for IP AH in Transport mode and Tunnel mode. Note how the IP AH header is inserted between the IP Header (IPH1) and the upper header (e.g. TCP Header) in Transport mode. Note also, in the tunnel mode the IPSec AH constructs another IP Header (IPH2) for the IP address of a destination gateway.

The ICV is computed first at the transmitter by the use of a common authentication algorithm that is also known to the receiver. Then ICV is recomputed at receiver and compared to match the received value for authentication integrity. ICV computation excludes non-predictable IP Header (IPH) fields like Time to live(TTL), Flags, Type of Service(TOS), Fragment offset, Checksum, etc.

0-7803-7400-2/02/$17.00 © 2002 IEEE

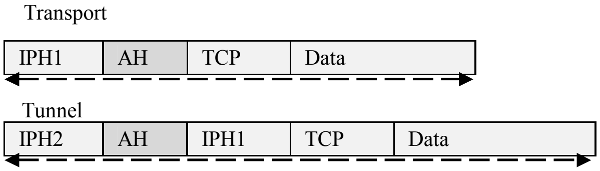

Figure 2. IP AH Authentication protection (dotted arrows).

If IP fragmentation occurs at the sender, it should be performed after AH processing. The IP reassembly should then be performed before AH processing at the receiver.

C. *IPSec Encapsulating Security Payload ESP*

Provides confidentiality (encryption), connectionless integrity (optional, not enforced at receiver end), data origin authentication (optional, not enforced at receiver end), and anti-replay integrity.

Figure 3 depicts the IPSec ESP header format. The Next Header field exactly as in IPSec AH. The Pad Length contains the number of pad bytes inserted by the encryption algorithm. The Sequence Number field is used same way as in IPSec AH. Finally, the Authentication Data field (variable size, multiple of 32 bits) contains ICV for the encapsulated packet and the ESP header/trailer (not including the authentication data itself.)

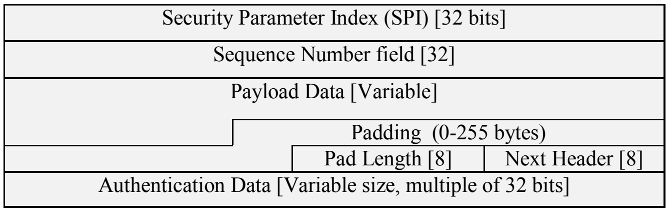
Security Parameter Index (SPI) [32 bits]
Sequence Number field [32]
Payload Data [Variable]
Padding (0-255 bytes)
Pad Length [8]
Next Header [8]
Authentication Data [Variable size, multiple of 32 bits]

Figure 3. IPSec ESP Header format.

Figure 4 depicts the protection range of authentication and encryption Transport mode and Tunnel mode. The ESP header is inserted exactly same way as in AH header. The ESP trailer is inserted after the payload data and before the ESP authentication data.

The ICV computation steps are the same as in IPSec AH.

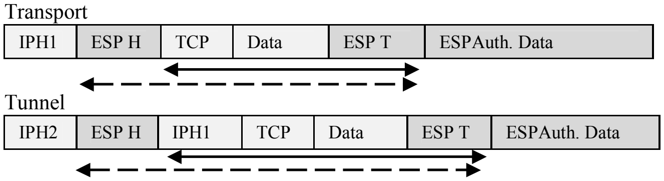

Figure 4. IP ESP Authentication protection (dotted arrows), and Encryption protection (solid arrows).

III. ANALYTICAL ANALYSIS

This section focuses on IPSec authentication and encryption space complexity, computation time complexity, and protocol limitations.

A. *Space comple ity.*

IPSec AH/ESP header size is 12/10 bytes fixed header fields respectively, plus the variable size authentication data. Authentication data is algorithm and packet specific field. IPSec authentication uses Keyed-Hashing for Message Authentication (HMAC) [8] combined with Message Digest algorithm (MD5) [9], or Security Hash Algorithm (SHA-1) [10]. All algorithms use secret key distribution. In combined HMAC-MD5, the key size is 128 bit while in the combined HMAC-SHA-1 the key size is 160 bits. In both cases the algorithm works with 64-byte message blocks. They generate a truncated ICV of 96 bits (12 bytes) to conform with the IPSec AH and ESP authentication data size. This means a total header size of 24/22 bytes for AH/ESP respectively, per packet. This overhead is needed for each authentication request transmitted through a security association in transport mode. Additional 20 bytes of IP header are also needed per packet for tunnel destination in tunnel mode. A total overhead of 44 bytes per packet assumes no header compression mechanisms used. Table 1 summarizes these results for both Transport and Tunnel modes.

TABLE I
IPSEC HEADER FIELD SIZES (BYTES)

Protocol	IPSec AH			IPSec ESP		
	Fixed	Variable	*Total*	Fixed	Variable	*Total*
Transport Mode	12	12	*24*	10	12	*22*
Tunnel Mode	12 20	12	*44*	10 20	12	*42*

IPSec ESP encryption may use extra padding bytes (0 to 255 bytes). They are algorithm and payload specific.

B. *Time comple ity.*

1 Encryption IPSec ESP Encryption algorithm is triple Data Encryption Standard (3DES) in Cipher Block Chaining (CBC) mode [11]. Encryption using DES algorithm is the most time consuming process. DES uses a 56-bit key, and block sizes of 64 bits. The algorithm has 19 distinct steps. The first step is a key independent transposition on the 64bit input block. The last step is the exact inverse of this transposition. In step 18, a 32 bit SWAP operation is performed. The remaining steps (2 to 17) are functionally identical and are dependent on different portions of the input key. Each of these 16 steps takes two 32 bit inputs, and produce two 32 bit outputs. The left output is a COPY of right input. The right output is an XOR of left input and a function *f* of right input and the step key. The function *f* consists of 4 operations. First, a 32:48 bit transposition/expansion of the right input is applied; second a 48 bit XOR of the output with the step key. Then a group mapping is performed to reduce the output size from 48 to 32 bits using two dimensional look-up table of 4 rows by 16 columns for all possible 64 inputs. Finally, 32-bit transposition is performed.

Each of the 16 steps has a special key, which is derived from the 56-bit key using the *KS* function. For the *KS* function, an initial 56-bit transposition is performed to the input key. Before each step, the key is divided into two 28-bit sub-keys. Each of which is rotated left using LEFT SHIFT operation. Finally, 56:48 bit transposition/reduction is performed.

Table II lists DES basic operations and their equivalent simple ones. It also gives space requirement for each operation. Table III lists the DES and 3DES computations of total number of simple operations needed. 3DES is the chained form of DES with a chain size of 3. For simplicity, 3DES is assumed to have only 3 times number of operations as DES has. 3DES requires an extra random Initialization Vector (IV) of 8 bytes. For complete details of DES and 3DES algorithms see [12] [13].

TABLE II
DES BASIC OPERATIONS

Basic Operation	Equivalent simple operations		
	Type	Times needed	Space needed
b bit transposition	One dimensional table look-up	b	b
Two dimensional table map (for 6:4 bit map.)	Multiply	1	
	Add	1	
	One dimensional table look-up	1	4 rows x 16 cols.

As show in table III, the total number of operations per 64-bit block is 8091 operations. Given a packet size of N bits then the number of blocks is given by

$$n = \left\lceil \frac{N}{64} \right\rceil, \tag{1}$$

where $\lceil . \rceil$ means the smallest integer bigger than or equal to the operand. Therefore, 3DES time complexity is of $O\ \ n$.

TABLE III
DES OPERATIONS IN ONE BLOCK ENCRYPTION

Step	Operation	Times	Equiv. Total*	Notes
1,19	64 bit transposition	2	64x2	
2-17	32 bit COPY	16	16	16 steps
2-17	32 bit XOR	16	16	
2-17	48 bit transposition	16	48x16	*f* function
2-17	48 bit XOR	16	16	
2-17	6:4 bit Two dimensional Table Mapping	8x16	3x128	
2-17	32 bit transposition	16	32x16	
1	56 bit transposition	1	56	*KS* function
2-17	28 bit LEFT SHIFT	2x16	32	
2-17	48 bit transposition	16	48x16	
18	32 bit SWAP	1	1	Pre-output
			2697	DES Total
			8091	3DES Total

* Using equivalent simple substitutions from table II

Figure 5 shows the computed encryption time in micro seconds as function of input packet size in blocks of 64 bits

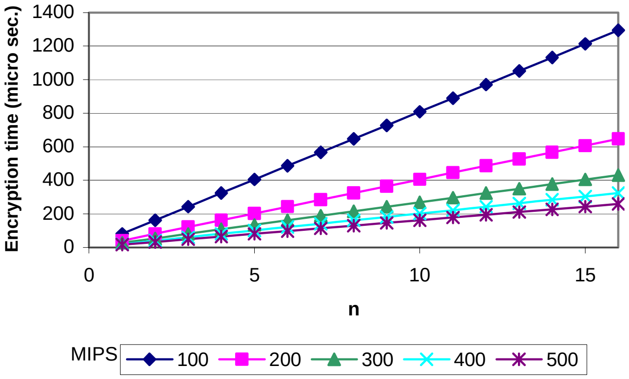

Figure 5. 3DES Encryption time vs. of input blocks and processing power.

and processing power in Millions of Instructions Per Second (MIPS). For example, time complexity of 3DES means approximately 4 Mbps encryption rate with a 500 MIPS.

2 *Authentication* As seen in previous section, IPSec Authentication algorithms are HMAC-MD5, or HMAC-SHA-1.

2-1 *MD5*

The first step in MD5 algorithm is padding the original message by appending 1 to 512 padding bits to it. The original message size is also appended in 64 bits. The total padded message with the message size field should have a size, which is multiple of 512 bits (64 bytes).

The algorithm uses four 4-byte registers (A, B, C, and D) which keep intermediate and, ultimately, the final value of the message digest of 128 bits. For each block of 64 byte of the message the algorithm performs 4 rounds of 16 steps each. In each step, one of the functions in table IV is computed in terms the registers (A, B, C, or D), the current input block, and a lookup value. By investigating MD5, one can find out that 10 to 12 operations per step are needed. Thus, in table IV, the total number of operations (T) is shown as 744 (720 24) operations per block. The time complexity is $O\ \ n$, where *n* is number of input blocks given by

$$n = \frac{N}{512} \quad ,\text{where} \tag{2}$$

$$N = \quad + pad + size . \tag{3}$$

is the input text, *pad* is the padding field, *size* is the size field, and *N* is the total message size.

Figure 6 shows the computed algorithm computation time (digest time) in micro seconds as function of input size in blocks of 512 bits (64 bytes) and processing power in MIPS. The algorithm has been tested on a Pentium II machine with 300 MHz rate for one million 1024 byte messages. The digest time computed was 159 seconds. The digest rate then is 51.5 Mbps. This means that every operation counted in table IV executed on the average in 6 cycles.

TABLE IV
MD5 OPERATIONS PER BLOCK

Round	Base Function ♦	Steps	Operations per step	*Total*
1	F(X,Y,Z)= X Y ∨ ¬X Z	16	*12*	*192*
2	G(X,Y,Z)= X Z ∨ Y ¬Z	16	*12*	*192*
3	H(X,Y,Z)= X ⊕ Y ⊕ Z	16	*10*	*160*
4	I(X,Y,Z)= Y ⊕ (X∨ ¬Z)	16	*11*	*176*
			Total (T)	*720*

♦ Symbol for AND, ∨ for OR, ¬ for NOT, and ⊕ for EXCLUSIVE OR.
* Plus 24 operations per block for initialization and termination.

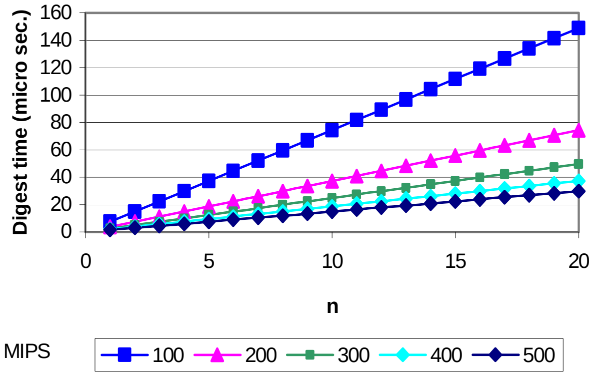

Figure 6. MD5 Digest time vs. of input blocks and processing power.

2-2 *HMAC-MD5*

The combined HMAC MD5 algorithm is formulated as follows:

$$MD5(K_o, MD5(K_i, Te\ \ t)) \quad \text{where}$$

$$K_i = \ Key \oplus ipad \qquad (4)$$

$$K_o = Key \oplus opad \qquad (5$$

K_i and K_o are two extended forms (512-bit) of the input *Key* and are generated by exclusive or the *Key* with *ipad* the inner padding (512 bits), and *opad* the outer padding (512 bits). *Key* is an arbitrary size secret key shared by sender and receiver. *Te t* is the given input message subject to authentication. For an input text of size bits, the number of input blocks for the inner MD5, n_k , is

$$n_k = \frac{N+K}{512} \quad ,\text{or} \qquad (6)$$

$$n_k = 1 + \frac{N}{512} \quad , \qquad (7)$$

where K is the size of the extra appended inner form of the key (512 bits).

In the outer MD5, the output of the inner MD5 (128-bit digest) is appended to K_o (512 bit). According to MD5, this is padded to two 512-bit blocks.

Thus, the HMAC-MD5 total number of operations (T) is

$$T(n_k) = 32 + (2 + n_k) \times 744 \, , \qquad (8)$$

where 32 comes since in (4),(5), the XOR operands are of size 512 bits. In a machine of word size 32 bits, this has to be partitioned in 16 consecutive XOR operations. A total of 32 extra operations is added. The time complexity is then $O\ n_k$.

Figure 7 shows the computed authentication time in micro seconds vs. of input blocks and processing power in MIPS.

2-3 *SHA-1*

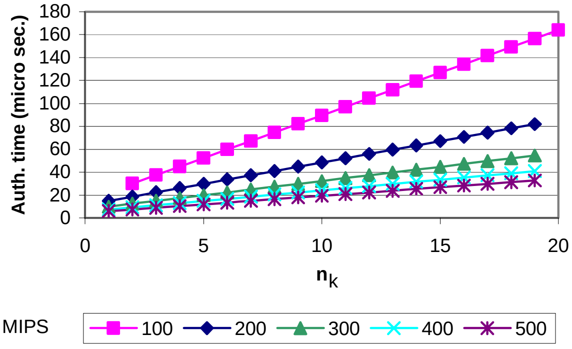

Figure 7.HMAC MD5 Authentication time vs. of input blocks and processing power.

SHA1 follows exactly the same way of message padding as in MD5 algorithm. However, the algorithm uses five 4-byte intermediate registers instead of four. Thus, the final value of the message digest is 160 bits. For each block of 64 bytes of the message the algorithm performs 4 rounds of 20 steps each. In each step, a functional computation based on the temporary registers, the current input block, and a constant value. By investigating SHA1, one can find out that 10 to 13 operations per step are needed. In table V, the total number of operations (T) is computed as 1110 (=900 210) operations per block. The algorithm s time complexity is $O\ \ n_k$, where n_k is the number of input blocks as in (6).

TABLE V
SHA1 OPERATIONS PER BLOCK

Round	Base Function	Steps	*Operations/ Step*	*Total*
1	F(X,Y,Z)= X Y ∨ ¬ X Z	20	*12*	*240*
2	G(X,Y,Z)= X ⊕ Y ⊕ Z	20	*10*	*200*
3	H(X,Y,Z)= X Y ∨ X Z ∨ Y Z	20	*13*	*260*
4	I(X,Y,Z)= X ⊕ Y ⊕ Z	20	*10*	*200*
			Total (T)	*900*

* Plus 210 operations per block for initialization and termination.

Figure 8 shows the computed digest time in micro seconds as function of number of input blocks and MIPS.

2-4 *HMAC-SHA-1*

The combined HMAC SHA1 algorithm is formulated as follows:

$$SHA1(K_o, \mathrm{SHA1}(K_i, Te\ \ t)) .$$

The total number of operations (T) needed for HMAC-SHA1 is of $O\ n_k$ where,

$$T(n_k) = 32 + (2 + n_k) \times 1110 \, . \qquad (9)$$

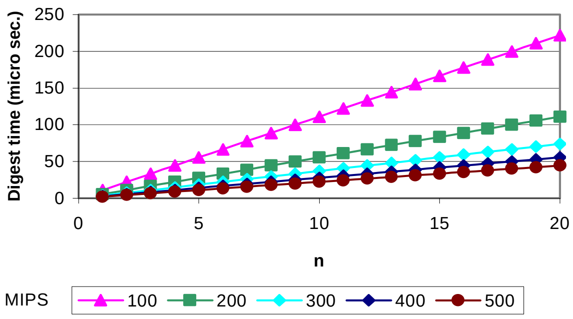

Figure 8. SHA1 digest time vs. of input blocks and processing power.

Figure 9 shows the computed authentication time in microseconds as function of of input blocks and processing power in MIPS.

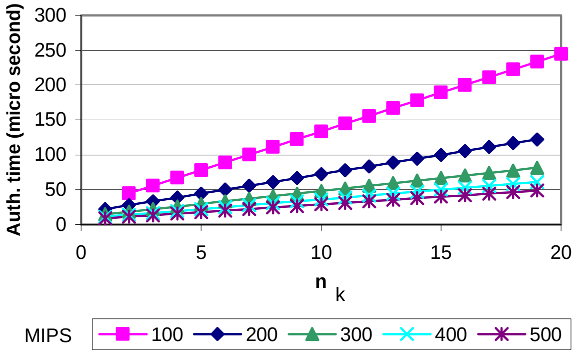

Figure 9. HMAC-SHA1 Authentication time vs. of input blocks and processing power.

In summary, Figure 10 shows a comparison between 3DES, HMAC-MD5, and HMAC-SHA1 throughput vs. processing power. It shows how HMAC MD5 supercedes HMAC SHA1. Also it shows how far the authentication throughput is as compared to encryption throughput.

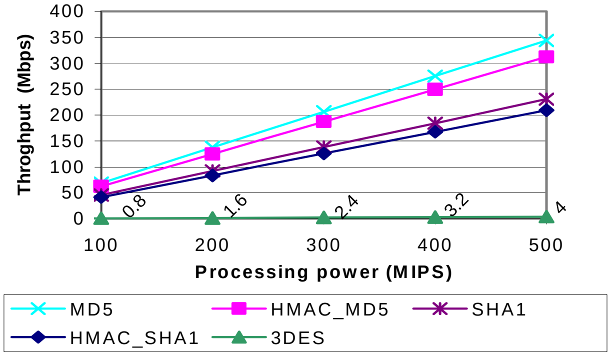

Figure 10. Throughput comparison vs. processing power.

C. Protocol limitations.

1) IPSec is IP based only. It does not provide security for any other network layer protocols like IPX, ATM, or FR etc.

2) Encryption using DES is based on a secret key of 56-bit size. In today s equipment this is very small key to break even if using brute force approach given the knowledge of plain text- cipher text pair. The addition of extra chaining of 3 levels (3DES) improves this limitation by the addition of an extra Initial Vector (IV) of random 64 bits. Practically, the IV can be set as the last 8 bytes of an encrypted packet for the next encryption process. In this case 3DES can be logically extended over consecutive packets [11]. Encryption should not be offered without the use of data origin authentication. After all, 3DES is much better than sending plain text.

IV. Summary and Conclusion

MD5 and SHA1 are both one-way hash functions that could be used with HMAC for IPSec authentication. MD5 is having higher throughput as compared to SHA1. Since the IPSec requires only 96 bits of the message digest, MD5 can be sufficient for the authentication purpose. 3DES encryption throughput is very low compared to authentication throughput. It should only be used in critical user information not for regular traffic flow. Encryption if needed should be combined with authentication. In this case if the message fails authentication, decryption process is saved (not performed).

References

[1] S. Kent, and R. Atkinson, IP Authentication Header, IETF RFC 2402,1998

[2] S. Kent, and R. Atkinson, IP Encapsulating security Payload (ESP), IETF RFC 2406,1998

[3] S. Kent, and R. Atkinson, Security Architecture for the Internet Protocol, IETF RFC 2401,1998

[4] A. S. Tanenbaum, *Computer Networks*, Prentice Hall PTR, 3rd Ed. PP 588-595, 1996.

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

[6] C. Madson, and R. Glenn, The Use of HMAC-SHA-1-96 within ESP and AH, RFC 2404, November1998

[7] O.Elkeelany, M.Matalgah, K.Sheikh, M. Thaker, D.Medhi, J. Qaddour, Layer 3 security architecture for portable MMDS Broadband wireless Network, Technical report, 8/2001.

[8] H. Krawczyk, M. Bellare, and R. Canetti, HMAC: Keyed-Hashing for Message Authentication, RFC 2104, February 1997.

[9] R. Rivest, The MD5 Message-Digest Algorithm, RFC 1321, April 1992.

[10] NIST, FIPS PUB 180-1: Secure Hash Standard, 1995. http://csrc.nist.gov/publications/fips/fips180-1/fips180-1.pdf

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

[12] US National Bureau of Standards, Data Encryption Standard, Federal Information Processing Standard (FIPS) publication 46-2, December 1993. http://www.itl.nist.gov/fipspubs/fip46-2.htm

[13] US National Bureau of Standards, DES modes of operation, Federal Information Processing Standard (FIPS) publication 81, December 1980. http://www.itl.nist.gov/fipspubs/fip81.htm