Performance Comparison of Message Authentication Code (MAC) Algorithms for the Internet Protocol Security (IPSEC).

Janaka Deepakumara, Howard M. Heys and R. Venkatesan
Electrical and Computer Engineering
Memorial University of Newfoundland
St. John's, NL, Canada, A1B3S7
Email: {janaka, howard, venky}@engr.mun.ca

ABSTRACT

The cryptographic algorithms employed in Internet security must be able to handle packets which may vary in size over a large range. Most of the cryptographic hash algorithms process messages by partitioning them into large blocks. Due to this fact the messages have to be prepared by padding the required amount of zero bits to get an integer number of blocks. This process contributes a considerable overhead when the short messages are more dominant in the message stream. Hashed Message Authentication Code-Secure Hash Algorithm-1 (HMAC-SHA-1) [1] has been recommended for message authentication in several network security protocols. The MAC based block cipher CBC-MAC-DES [2] has been included in the international standards for data integrity and authentication. However, after selecting the Advanced Encryption Standard (AES) algorithm, the use of DES merits reevaluation as Rijndael [3] shows good performance in both hardware and software and it has better security features than DES. CBC-MAC is likely to be standardized as an AES mode of operation. In this paper we will analyze the hardware and software performance of Hashed Message Authentication Code (HMAC) and Cipher Block Chaining Message Authentication Code (CBC-MAC) in the context of the traffic characteristics of the Internet. Studying the behavior of IP packet size of the Internet messages allows the estimation of actual performance of these authentication functions. The probability density function (PDF) of the IP packet size is approximated as one of four models, each having different accuracy level. The PDF is then used to determine the rate at which authentication can be executed on average, based on the results of previous hardware and software implementation performance data for the hash and encryption algorithms.

1. INTRODUCTION

Today the Internet has virtually become the way of doing business as it offers a powerful ubiquitous medium of commerce and enables greater connectivity of disparate groups throughout the world. However this medium has its inherent risks. Loss of privacy, loss of data integrity, identify spoofing and denial of service are some of the major threats in the Internet. Internet Protocol Security (IPSEC) addresses some of these issues by providing security services such as confidentiality, data integrity and authentication. The cryptographic algorithms employed in these services must be able to handle packets which may vary in size over a large range. The size of the message has a significant impact on the performance of such algorithms. In particular, the message authentication algorithms process messages partitioned into blocks. Hence the messages have to be prepared by padding the required amount of zero bits to get an integer number of blocks. This process becomes a considerable overhead when the short messages are more dominant in the message stream.

2. MESSAGE AUTHENTICATION IN IPSEC

IPSEC uses Authentication header (AH) to provide connectionless integrity, data origin authentication and an optional anti-replay service for IP datagrams [4]. This consists of the authentication data which is a variable-length field that contains the integrity check value (ICV) for this packet. The algorithms employed for ICV calculation are specified by a security association of IPSEC. For point-to-point communication keyed message authentication codes (MACs) based on symmetric encryption algorithms (e.g. DES, Rijndael) or on one way hash functions are used. For multicast communication one way hash algorithms combined with asymmetric signature algorithms are utilized [4]. Hashed based message authentication code (HMAC) has been the mandatory-to-implement MAC for IPSEC. HMAC based on secure hash algorithm (HMAC-SHA-1) has been recommended for message authentication in several network security protocols. The key reasons behind this are the free availability, flexibility of changing the hash function and reasonable speed, among others. The MAC based on the block ciphers such as CBC-MAC-DES was generally considered slow due to the complexity of the encryption process. As well, since DES has been shown to be not secure enough against the growing computing power it is not recommended any

more. However, after selecting the AES encryption algorithm, this situation merits reevaluation as Rijndael [3] shows good performance in both hardware and software and it has better security features than DES. CBC-MAC is likely to be standardized as an AES mode of operation.

2.1 Hashed Based Message Authentication Code

HMAC proposed by M. Bellare et al. in 1996 [5] has been standardized in 2001 [1]. The main objective of HMAC is the use with no modification, of available hash functions, which perform well and for which software codes are freely available. HMAC can be proven secure if the embedded hash function has a reasonable cryptographic strength. Let H be the hash function initialized with its fixed Initial Value, IV, that results in an n-bit hash value. The HMAC works on inputs M of arbitrary length, which is a multiple of b bits. It uses a single random string K as its key. If the key length is greater than b, it is input to the hash function to produce an n-bit key. The recommended key length is $\geq n$. The HMAC can be expressed as;

$$\mathrm{HMAC}_k(M) = H(K^+ \oplus opad, H(K^+ \oplus ipad, M))$$

where K^+ is K padded with zeros on the left so that the result is b bits long, *ipad* is the inner pad, which is the byte hex 36 repeated $b/8$ times and *opad* is the outer pad, which is the byte hex 5C repeated $b/8$ times. The symbol $\oplus$ denotes the XOR operation. In the federal information processing standard (FIPS) HMAC is specified for an arbitrary approved cryptographic hash function H. We assume a block size of $b = 512$ bits.

2.2 Cipher Block Chaining Message Authentication Code (CBC-MAC)

Let a message space for M be binary strings whose lengths are a positive multiple of l. Hence the message M can be broken into blocks such that

$$M = M_1, M_2, .., M_m \text{ with } \quad |M_i| = l.$$

Then each block is passed through the encryption E with key K and the result is XORed with the next block. If E_K represents the encryption using a key K then cipher block chaining is given by

$$\mathrm{CBC}_{E_K}(M) = \sum_{i=1}^{m} E_K[M_i \oplus C_{i-1}] \qquad \text{For } i = 1 \ldots m \text{ and } C_0 = 0^l$$

where, 0^l indicates l bit vector of all zeros.

The CBC-MAC comes in different versions varying in details such as padding, length variability and key search strengthening [6]. The general way of padding for CBC-MAC is by considering the final input block as a partial block of data, left justified with zeroes appended to form a full block [2]. In this analysis for CBC-MAC analysis, we assume a block size of 128 bits (as is the case for AES).

3. PREVIOUS STUDIES OF INTERNET TRAFFIC

The statistical properties of Internet traffic are complex and the amount of data to be studied is very large. The understanding of Internet traffic is useful to study the performance of authentication algorithms. Various attempts have been made over the years to study the nature of the wide area Internet traffic.

Several important observations have been made by the analyses of actual traffic carried in the Internet backbone. The Cooperative Association for International Data Analysis (CAIDA) has performed studies on backbone traffic characteristics at a site inside a major Internet traffic exchange point over a period of 10 months [7]. The traces for the study were collected from an OC-3 ATM link using an optical splitter. The distribution of IP packet sizes is shown in Figure 1.

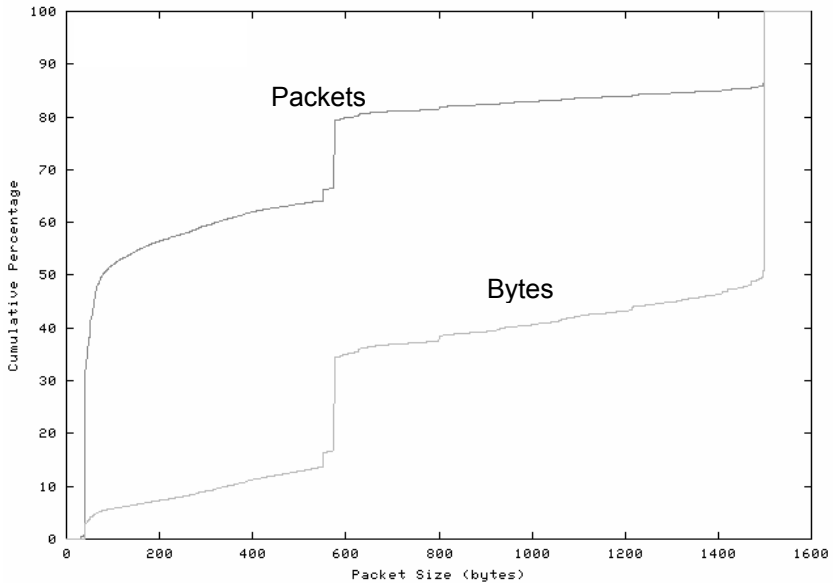

Figure 1. Cumulative distribution of IP packet sizes [8].

According to this study, TCP has contributed to about 85% of the traffic in the traces. A large portion of TCP traffic has been generated by bulk transfer applications such as HTTP and FTP. The packet size has been categorized into three groups namely 40 byte packets (minimum packet size for TCP), which carry TCP acknowledgments but no payload, 1500 byte packets (maximum Ethernet payload size) and 552 and 576 byte packets, from TCP.

A rule of thumb employed in some of the IP traffic analysis has three packet sizes each having equal probability of occurrence. They are 40 bytes, 256 bytes and 1500 bytes, each having a probability of occurrence of one third [9].

3.1. IP Packet Models

Studying the behavior of IP packet size of the Internet messages allow the estimation of actual performance of authentication functions such as HMAC-SHA-1 or CBC-MAC. The average block size can be derived by analyzing the statistical observations. For analytical purposes the probability density function (PDF) of the IP packet size will be approximated as one of the following four cases:

Case (i)

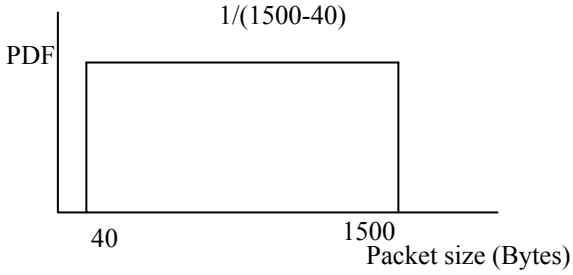

Figure 2. Uniform PDF

Case (ii)

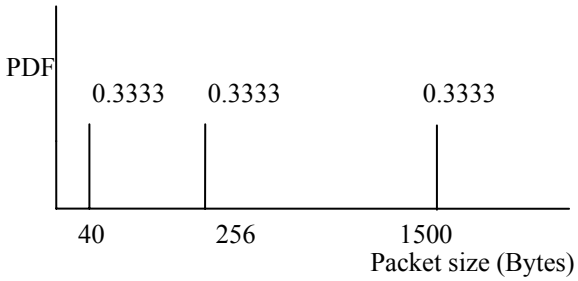

Figure 3. Rule-of- thumb PDF

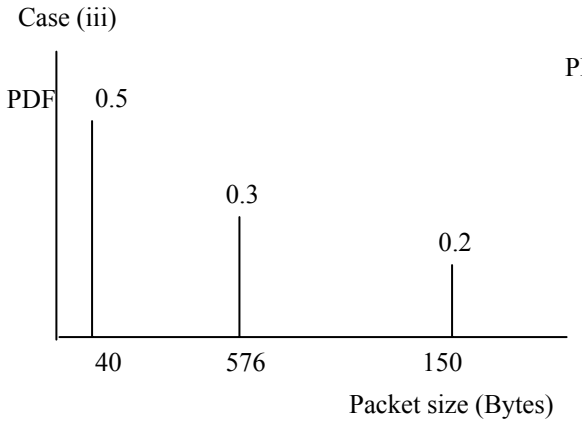

Figure 4. Discrete PDF with 3 impulses.

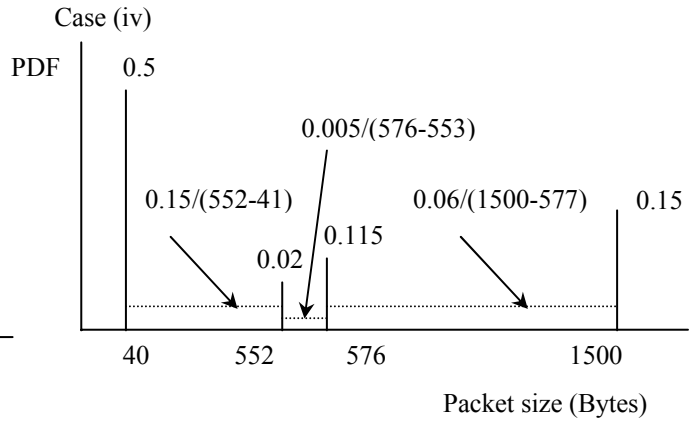

Figure 5. Discrete and Uniform PDF

Each case has a different degree of accuracy in modeling Internet traffic. Case (i) is the straightforward assumption of uniformly distributed packet sizes; Case (ii) is a typical rule-of-thumb; Case (iii) is a better rule-of-thumb; Case (iv) follows most closely Figure 1.

The average number of blocks per packet (Γ) can be obtained as follows. If the number of blocks of the message of packet size i bytes is $Nblk(i)$ and the probability of the message has i bytes is $p(i)$,

$$\Gamma = \sum_{i=40}^{1500} p(i) \times Nblk(i) \qquad (4.1)$$

4. PERFORMANCE OF MACS IN INTERNET

Let us take the number of blocks of the message of packet size i bytes for HMAC and CBC-MAC using AES (CBC-MAC-AES) are $Nblk_{MAC}$ and $Nblk_{AES}$, respectively. According to the padding method described in HMAC-SHA-1 [1], $Nblk_{MAC}$ can be obtained as follows:

$$Nblk_{MAC} = \begin{cases} \lceil (i \times 8)/512 \rceil + 1 & \text{if } (i \times 8) \bmod 512 = 0 \text{ or } (i \times 8) \bmod 512 \geq 448 \\ \lceil (i \times 8)/512 \rceil & \text{if } (i \times 8) \bmod 512 < 448 \end{cases}$$

Assuming the general way of padding described above has been adopted for CBC-MAC-AES,

$$Nblk_{AES} = \lceil (i \times 8)/128 \rceil.$$

4.1 Average Number of Blocks per Packet

The average number of blocks per packet of the IP traffic can be estimated according to the four cases mentioned earlier.

Case (i):

In this case the PDF of the packet size is assumed to be uniformly distributed. From Figure 2, the average number of blocks for HMAC can be estimated from equation (4.1), as follows;

HMAC	CMC-MAC-AES
$\Gamma = \sum_{i=40}^{1500} [(0.5/1460) \cdot Nblk_{MAC}(i)]$	$\Gamma = \sum_{i=40}^{1500} [(0.5/1460) \cdot Nblk_{AES}(i)]$
$\Gamma = 6.34$	$\Gamma = 24.34$

Case (ii):
This is a rule of thumb used in some performance analyses related to IP traffic. From Figure 3, with equation 4.1, the average number of blocks for HMAC and CBC-MAC can be estimated based on following calculations:

HMAC	CMC-MAC-AES
$Nblk_{MAC}(40) = 1$	$Nblk_{AES}(40) = 3$
$Nblk_{MAC}(256) = 5$	$Nblk_{AES}(256) = 17$
$Nblk_{MAC}(1500) = 24$	$Nblk_{AES}(1500) = 94$
$\Gamma = 10$	$\Gamma = 38$

Case (iii):
This model has a PDF with three impulses as shown in Figure 4. With equation 4.1, the average number of blocks for HMAC and CBC-MAC can be calculated as follows:

HMAC	CMC-MAC-AES
$Nblk_{MAC}(40) = 1$	$Nblk_{AES}(40) = 3$
$Nblk_{MAC}(576) = 10$	$Nblk_{AES}(576) = 37$
$Nblk_{MAC}(1500) = 24$	$Nblk_{AES}(1500) = 94$
$\Gamma = 8.3$	$\Gamma = 31.4$

Case (iv):
This model closely follows the cumulative distribution given in Figure 1. From Figure 5 the average number of blocks for HMAC and CBC-MAC can be estimated using:

HMAC	CMC-MAC-AES
$Nblk_{MAC}(40) = 1$	$Nblk_{AES}(40) = 3$
$Nblk_{MAC}(552) = 9$	$Nblk_{AES}(552) = 35$
$Nblk_{MAC}(576) = 10$	$Nblk_{AES}(576) = 37$
$Nblk_{MAC}(1500) = 24$	$Nblk_{AES}(1500) = 94$

From equation (4.1), the average number of HMAC blocks is found as follows:

$$\Gamma = (0.5)(1) + \sum_{i=41}^{551}[(0.15/511)\cdot Nblk_{MAC}(i)] + (0.02)(9) + \sum_{i=553}^{575}[(0.005/23)\cdot Nblk_{MAC}(i)]$$
$$+ (0.115)(10) + \sum_{i=577}^{1499}(0.06/923)\cdot Nblk_{MAC}(i) + (0.15)(24)$$

where the summations of the number of blocks can be easily found as shown below:

$$\sum_{i=41}^{551} Nblk_{MAC}(i) = 2687\ ,\quad \sum_{i=553}^{575} Nblk_{MAC}(i) = 215\ ,\quad \sum_{i=577}^{1499} Nblk_{MAC}(i) = 15558\ .$$

Therefore, the average number of HMAC blocks with this model, which closely follows the actual packet distribution is used for the IP packet size, is $\Gamma = 7.28$.
From Figure 5 and equation (4.1) the average number of CBC-MAC blocks is given by

$$\Gamma = (0.5)(3) + \sum_{41}^{551}[(0.15/511)\cdot Nblk_{AES}(i)] + (0.02)(35) + \sum_{553}^{575}[(0.005/23)\cdot Nblk_{AES}(i)].$$

Hence, the average number of CBC-MAC blocks for the model, which closely follows the actual packet distribution, is 27.51.

4.2 Performance in Hardware

The performance of the FPGA implementations of HMAC-SHA-1 (using full loop unrolled results of SHA-1) [10] and CBC-MAC-AES is compared in the four IP traffic cases previously mentioned. The FPGA implementation results of Rijndael published by NIST have been used for this analysis. An FPGA implementation of Rijndael has been carried out using Xilinx Virtex XCV1000BG560-4 device by A.J Elbirt et al. [11] for the evaluation of AES finalist algorithms. The clock frequency that has been obtained by them for the loop unrolled architecture in feedback mode is 14.1 MHz. One block has taken 6 clock cycles and thus a throughput of 300.1 Mbps has been obtained. The delay for one block encryption is 426.5 ns. However, for comparison purposes the

speed grade of the Virtex device has to be taken as –6 as with our implementations. The change of speed grade –4 to –6 gives approximately 28% of speed enhancement [12]. Hence, the delay for block encryption would approximately be 307.08 ns. The average time for calculating MAC using HMAC and CBC-MAC-AES can be determined by the following relationships:

$$\text{Average HMAC calculation time} = (3 + \Gamma) \times [\text{time for a hash}] \quad (4.2)$$

$$\text{Average CBC-MAC calculation time} = \Gamma \times [\text{time for a block encryption}] \quad (4.3)$$

The calculated values for HMAC and CBC-MAC-AES using equations (4.2) and (4.3) are given in Table 1.

Table 1. Times for HMAC-SHA-1 and CBC-MAC-AES on FPGA for general IP traffic

	Case (i)		Case (ii)		Case (iii)		Case (iv)	
	Γ	Average packet time(μs)	Γ	Average packet time(μs)	Γ	Average packet time(μs)	Γ	Average packet time(μs)
HMAC-SHA-1	6.34	8.21	10	11.71	8.3	10.18	7.28	9.25
CBC-MAC-AES	24.34	7.48	38	11.67	31.4	9.64	27.51	8.45

4.3 Performance in Software

Both HMAC-SHA-1 and CBC-MAC-AES were run on a 927 MHz Pentium II machine. The C code which has been used for assessing the speed performance of Rijndael by NIST [13] was used in CBC mode for CBC-MAC-AES. The plaintext of 128-bit was encrypted 1,000,000 times in CBC mode using a 128-bit key. The average time for encrypting a 128-bit block was 1.14 μs. The C code of SHA-1 published by DI management [14] according to NIST specifications was used to determine the software speed of hashing. The average time taken for creating a hash value for a 512-bit block was 3.001 μs.

Table 2. Times for HMAC-SHA-1 and CBC-MAC-AES on software for general IP traffic

	Case (i)		Case (ii)		Case (iii)		Case (iv)	
	Γ	Average packet time(μs)	Γ	Average packet time(μs)	Γ	Average packet time(μs)	Γ	Average packet time(μs)
HMAC-SHA-1	6.34	28.03	10	39.01	8.3	33.91	7.28	30.84
CBC-MAC-AES	24.34	27.75	38	43.32	31.4	35.80	27.51	31.36

5. CONCLUSION

According to Table 1, the average times of MAC calculations of the FPGA implementations of HMAC and CBC-MAC-AES do not differ significantly. Particularly, they are almost the same as the size of the packets becomes larger. The software results given in Table 2 show that the timing performances of HMAC-SHA-1 and CBC-MAC-AES do not have a significant difference. As observed in FPGA implementations HMAC performance improves significantly as the packets become larger. In general, the size of the packet has a considerable impact on the performance of cryptographic hash algorithms as the padding has to be carried out all the time even if the message has a length of multiples of 512 bits. This becomes a large overhead for small packets especially in case of HMAC algorithm. Unfortunately most of today's cryptographic hash algorithms follow a sequential structure, which is difficult to pipeline. Many of them cannot meet today's speed requirements. Hence, new approaches of hash algorithms, which are more efficient in speed, have to be explored.

REFERENCES

[1] FIPS PUB # HMAC, " *The Keyed Hash Message Authentication Code (HMAC)*," Federal Information Processing Standard (FIPS) Publication # HMAC, National Institute of Standards and Technology, US Department of Commerce, Washington D.C., 2001.

[2] FIPS 113, " Computer Data Authentication," Federal Information Processing Standard (FIPS), Publication 113, National Institute of Standards and Technology, US Department of Commerce, Washington D.C., 1985.

[3] FIPS PUB 197, "Advanced Encryption Standards (AES)," Federal Information Processing Standard (FIPS), Publication AES Draft, National Institute of Standards and Technology, US Department of Commerce, Washington D.C., 2001

[4] S. Kent and R. Atkinson, " *IP Authentication Header*" IETF Network Working Group, RFC 2402, November 1998.

[5] M. Bellare, R. Canetti and H. Krawczyk, "Keying Hash Functions for Message Authentication," in *proceedings of Advances in Cryptology- CRYPTO'96*, Lecture Notes in Computer Science Vol. 1109, Springer-Verlag, pp. 1-15, 1996.

[6] J. Black and P. Rogaway, " CBC MACs for Arbitrary-Length Messages: The Three-Key Constructions," in proceedings of advances in Cryptology-CRYPTO'00, Lecture Notes in Computer Science, Springer-Verlag vol. 1880, pp. 197-215, 2000.

[7] A. Feldmann, J. Rexford and R. Caceres, "Efficient Policies for Carrying Web Traffic Over Flow-Switched Networks," *IEEE/ACM transactions on Networking*, pp. 673-685, December 1998.

[8] S. McCreary and K. Claffy, "Treds in Wide Area IP Traffic Patterns," A view from Ames International Exchange, Cooperative Association for International Data Analysis (CAIDA) Report, 2000, available at http://www.caida.org/ outreach/ papers/AIX0005.

[9] J. Black, S. Halevi, H. Krawczyk, T. Kovetz and P. Rogaway, "Update on UMAC Fast Message Authentication" available at http://www.cs.ucdavis.edu/~rogaway/ umac.

[10] J. Deepakumara, H. M. Heys and R. Venkatesan, "*Performance of FPGA implementation of Hashed Message Authentication Code-Secure Hash Algorithm (HMAC-SHA)*", Newfoundland Electrical and Computer Engineering Conference, (NECEC) 2001, November 13, 2001.

[11] A. Elbirt, W. Yip, B. Chetwynd and C. Paar, "An FPGA Implementation and Performance Evaluation of the AES Block Cipher Candidate Algorithm Finalists," in *proceedings of 3rd AES Candidate Conference*, available at http://www.nist.gov/aes.

[12] Xilinx home page: http://www.xilinx.com.

[13] NIST home page, "AES Algorithm (Rijndael) Information", available at http://csrc.nist.gov/encryption/aes/ rijndael/.

[14] "Sha1.c: Implementation of the Secure Hash Algorithm", November 2000, available at http://www.di-mgt.com.au/src/sha1.c.txt.