IPSec Overhead in Wireline and Wireless Networks for Web and Email Applications

George C. Hadjichristofi Nathaniel J. Davis, IV Scott F. Midkiff
Bradley Department of Electrical and Computer Engineering
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061 USA

Abstract

This paper focuses on characterizing the overhead of IP Security (IPSec) for email and web applications using a set of test bed configurations. The different configurations are implemented using both wireline and wireless network links. The testing considers different combinations of authentication algorithms and authentication protocols. Authentication algorithms include Hashed Message Authentication Code-Message Digest 5 (HMAC-MD5) and Hashed Message Authentication Code-Secure Hash Algorithm 1 (HMAC-SHA1). Authentication protocols include Encapsulating Security Payload (ESP) and Authentication Header (AH) protocols. Triple Digital Encryption Standard (3DES) is used for encryption. Overhead is examined for scenarios using no encryption and no authentication, authentication and no encryption, and authentication and encryption. A variety of different file sizes are considered when measuring the overhead. The results present a thorough analysis of the overhead of different IPSec configurations and provide practical guidance for choosing the IPSec configuration needed in a network environment.

1. Introduction

Information on the Internet is carried using the Internet Protocol (IP), which does not inherently provide privacy or other security. As a result, IP Security (IPSec) was developed to integrate security into IP. IPSec provides connectionless data integrity, authentication, data confidentiality, anti-replay protection, data origin authentication, and limited traffic flow confidentiality.

Previous work by McGregor and Lee investigated the effects of combining data encryption with data compression [1]. The authentication and encryption algorithms were implemented in C and the throughput of each algorithm was measured by finding the required time for authenticating and/or encrypting files of various sizes. The results for the throughput of each algorithm were then inserted in a system model to calculate the overall speedup when compression was used in different types of networks. The research did not involve deploying test beds for "live" experimental measurements.

Chappell, *et al.* did additional work [2]. The purpose of their research was to determine the suitability of IPSec for a Multiple Level Security environment. The research used an experimental version of IPSec that was not optimized for performance. A simple IPSec wireline test bed was deployed and various measurements were made. The research did not investigate the overhead of using authentication and encryption and the IPSec implementation used only DES for confidentiality.

The results presented in this paper are derived from actual measurements taken on wired and wireless test beds and provide an expanded testing environment when compared to the work done in [1] and [2]. The overhead of authentication and encryption was investigated using the 3DES encryption algorithm. Data was collected using a greater range of file transfer sizes. Of particular significance, our experiments used a standard, commercial off-the-shelf (COTS) implementation of IPSec, Linux FreeS/WAN. Data gathered about the performance of the COTS implementation supports efforts to enhance network interoperability. Finally, the metrics of network load and number of transactions required (from server to client) give a different perspective of IPSec overhead since these metrics are independent of processing capabilities of the networked computers. Previous work measured throughput, which is inherently related to the computers' processing capabilities.

2. IPSec Architecture

IPSec is a suite of protocols, including AH, ESP, IKE, ISAKMP/Oakley and various transforms [3]. IPSec defines how these different components interact with each other to implement the required functionality. This research focuses on IPSec's tunnel mode used to provide

0-7803-7893-8/03/$17.00 © 2003 IEEE

subnet-based security to create a Virtual Private Network (VPN). It considers both the AH and ESP security protocols. The primary difference between AH and ESP authentication is the extent of coverage. ESP does not authenticate any IP header fields of the outer IP header. AH can provide a better check of integrity, if required, since it extends its protection to predictable fields of the outer IP header. However, using AH introduces extra overhead, which is investigated in this paper. The algorithms that provide authentication for the AH and ESP protocols are SHA1 and MD5. The algorithm that provides encryption for ESP is 3DES. The Security Policy is also an important part of a security architecture. It determines the transforms that two entities should use to communicate with each other. Based on the security policy, key management generates and manages a key by using the Internet Key Exchange (IKE).

3. Test Bed Configurations

A wireline test bed with two computers was deployed to determine the overhead for the different algorithms and different protocols (see Figure 1). A 10-Mbps Ethernet local area network was used to connect the two hosts. A third computer, "Sniffer," collected data. An IPSec tunnel connected computers "West" (client) and "East" (server).

To observe IPSec overhead over only a wireless link, test bed configuration 1 was modified to form test bed configuration 2 (see Figure 2). An IEEE 802.11 wireless local area network operating at 2 Mbps connected computer "East" to computer "West."

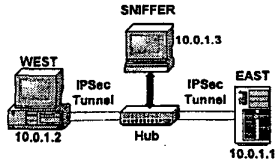

Figure 1. Test bed configuration 1.

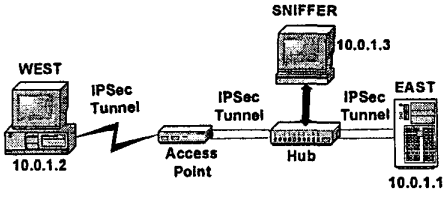

Figure 2. Test bed configuration 2.

One test bed variation examined the effects of a heterogeneous configuration of wireline and wireless media. Another variation investigated the effects of using clients with various processing capabilities. The processor for "East" was a 566-MHz Intel Celeron, the processor for "West," the slower client (SC), was a 200-MHz Pentium Pro, and the processor for "Dusk," the faster client (FC), was a 1-GHz Pentium III. Red Hat Linux was used as the operating system for all computers. FreeS/WAN IPSec [5] was installed on all computers except "Sniffer." Version 4 of the Internet Protocol (IPv4) was used for all scenarios.

The various combinations used for the experiments are shown in Figure 3. The different experimental scenarios are specified using the following notation.

{Number + (M)anual/(A)utomatic Keying}-
[Authentication Protocol-Authentication Algorithm]-
[Encryption Protocol-Encryption Algorithm]

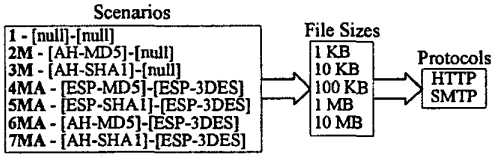

Figure 3. Combinations used for experiments.

The metrics recorded in each scenario were the network load in bytes per second, the number of transactions, and the time in seconds required to transfer a file from the server to the client. The network load and number of transactions were separated into client to server (CtoS) and server to client (StoC) traffic.

4. Analysis of Results

Data for the different experimental scenarios were aggregated to evaluate the behavior of IPSec with respect to protocols, algorithms, and file sizes. The different behaviors were evaluated by analyzing variations in the metrics recorded. When describing results, each scenario is referenced by the notation specified in Figure 3, e.g., 1, 2M, 3M, 4MA, etc.

4.1 Overhead with Wireline Transmission Links

This section describes the overhead of using no authentication and no encryption versus using authentication and no encryption or using both authentication and encryption. Test bed configuration 1 was used to produce two sets of data, one using a slower client computer ("West") and the other using a faster client computer ("Dusk"). The increase in overhead for securing data was calculated by finding the percentage increase in the network load, number of transactions, and transfer time for each scenario compared to the scenario using no authentication and no encryption (scenario 1). Due to space limitations, only data for SMTP that

includes the Ident protocol overhead are presented. (Sendmail used the Ident protocol at the beginning of an email transfer [4].)

The increase in the number of transactions for 1-KB and 10-KB files using the fast and slow clients was negligible for both HTTP and SMTP. Figure 4 shows results for HTTP using the slow client. For larger files and the fast client, the number of transactions increased by approximately 5% for scenarios 2M through 7MA using both HTTP and SMTP with the Ident protocol.

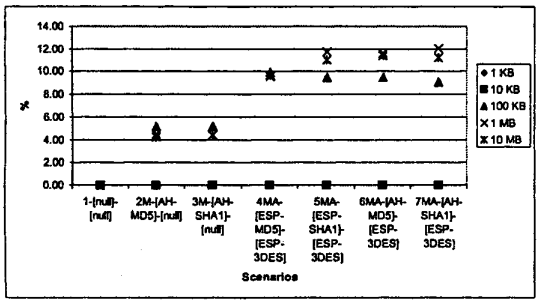

Figure 4. Increase in total transactions (HTTP, SC).

However, for the slow client and scenarios 4MA to 7MA, the increase in the number of transactions was 10% for HTTP and 22% for SMTP. This larger increase was due to an increase in the number of CtoS acknowledgements (ACKs) from the slower client computer, "West," that slowed the server. "West" did not have the computational capabilities to process packets at the rate at which the server could send them. The reason for the 22% increase for SMTP, compared to the 10% increase for HTTP, was because SMTP has higher computational requirements than HTTP. These higher computational requirements slowed down "West" even further, decreasing its processing rate for the received packets. As a result, more ACKs were required with SMTP than with HTTP to control the packet flow.

The percentage increase in the network load for CtoS and StoC for various file sizes and scenarios 2M and 7MA relative to using no authentication and no encryption (scenario 1) is shown in Figure 5 for HTTP using the slow client. The 2M scenario had the least increase in the network load and the 7MA scenario had the greatest increase. The increase in the network load for the CtoS case increased due to the number of ACKs sent to control the traffic flow. However, the increase in the network load for StoC case, for both the 2M and the 7MA scenarios, decreased as the file size increased. This was

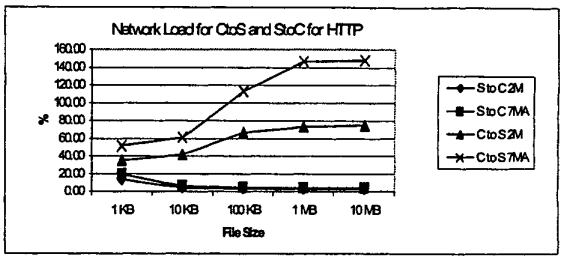

Figure 5. File size effect on load (HTTP, SC).

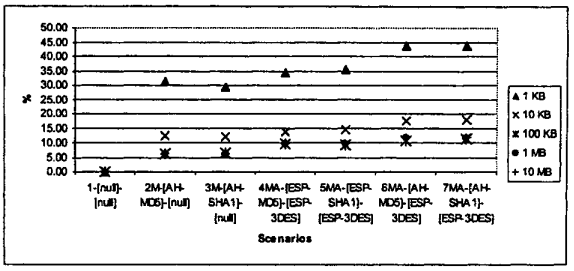

Figure 6. Increase in load (SMTP, SC).

because IPSec overhead constituted a smaller fraction of the overall load as the file size increased.

The general pattern for the increase in network load with respect to all scenarios for different file sizes is shown in Figure 6 for SMTP using the slow client. Both HTTP and SMTP demonstrated a similar pattern. Overall, as compared to using no authentication and no encryption (scenario 1), SMTP had a greater increase in network load than HTTP.

The ranges for the lowest and highest increase in network load, as a percentage, for test bed configuration 1 are shown in Table 1 for HTTP and SMTP with the fast and slow clients. The faster client yielded a smaller increase in the network load for the 10-MB file mainly due to the decrease in the CtoS ACKs sent.

Table 1. Increase in Network Load

Protocols	1 KB		10 MB	
	Lowest % -2M	Highest %-7MA	Lowest %-2M	Highest %-7MA
HTTP – SC	20	30	5	8
HTTP – FC	20	30	3	5
SMTP – SC	31	44	6	12
SMTP – FC	31	44	4	6

The graph in Figure 7 depicts how the transfer time for each file size varied for each scenario relative to scenario 1 for HTTP using the slow client. The basic relationship for the increase in transfer time among the various algorithms can be ordered as follows.

$$1 < 2M < 3M < 4MA < 6MA < 5MA < 7MA$$

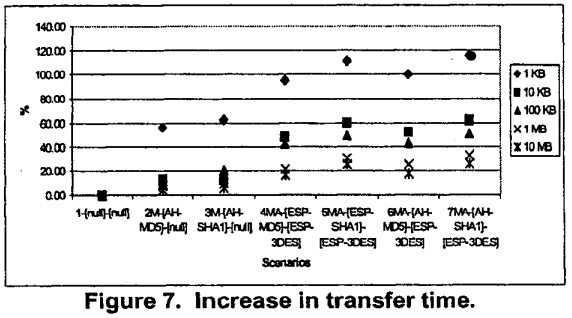

Figure 7. Increase in transfer time.

The relative increase in the transfer time for the smaller files was greater than the relative increase for the larger files, as shown in Table 2. When transferring larger files with the slow client, the transfer time was affected by the CtoS ACKs that were utilized to slow the packet flow. However, when transferring larger files, with the fast client, packet flow control was required only when using authentication and SMTP. Overall, the total transfer time for sending a file with SMTP was higher than with HTTP, thus yielding a smaller percentage increase in the transfer time for each scenario.

Table 2. Percentage Increase in Transfer Time

File Size	Scenarios	HTTP SC	HTTP FC	SMTP SC	SMTP FC
1 KB	Lowest % 2M	56	18	9	30
	Highest % 7MA	116	52	25	40
10 MB	Lowest % 2M	5(*)	3	14(*)	6(*)
	Highest % 7MA	26(*)	11	65(*)	1
(*) Affected by the increase in the CtoS ACKs					

4.2 Overhead with Wireless Transmission Links

This section investigates the impact of wireless links on IPSec overhead based on using test bed configuration 2. The analysis for comparing the overhead for different scenarios for test bed configuration 2 was conducted following the same approach used for test bed configuration 1. The results from test bed configuration 2 indicated that the patterns of overhead for wireless and wireline links were similar, at least for the environment used in these experiments.

The relative increase in the network load and number of transactions for test bed configuration 1, using the fast client, was approximately the same as for test bed configuration 2. Even though test bed configuration 2 was deployed with the slow client, "West," the relative increase more closely matched the results for the fast client of test bed configuration 1 because the wireless medium became the network bottleneck. The network nodes had more time and could process and prepare the data for transmission faster than it could be sent. As a result, no extra ACKs were needed to control the packet flow.

The increase in transfer time, as a percentage, for scenarios 1 through 7MA using test bed configuration 2 was less than the increase in transfer time using test bed configuration 1. Even though more time was required to send the additional overhead due to authentication and/or encryption over the wireless link, the time increase constituted a smaller part of the total time to transfer a file, thus yielding a lower relative increase in transfer time. Table 3 shows the percentage increase in transfer time for scenarios 2MA and 6MA when sending data over the wireless link. The Ident time was subtracted from the data. The overhead of Scenario 7MA was not measured to avoid redundancy.

Table 3. Percentage Increase for All Metrics

Metrics	Protocols	10 KB		1 MB	
		2M (%)	6MA (%)	2M (%)	6MA (%)
Transfer Time	HTTP	6	28	5	6
	SMTP	6	14	3	8

The mean and standard deviation of the ratio of the transfer time for each scenario using the different test beds are shown in Table 4. Specifically, the transfer time for a scenario using test bed configuration 2 (T2) was divided by the transfer time for the same scenario using test bed configuration 1 (T1), with the fast client or slow client, or the heterogeneous configuration (HC) with wireline and wireless media. The mean and standard deviation were then calculated based on the ratios of all the scenarios for each file size and protocol.

Table 4. Mean and Standard Deviation of the Ratios of Transfer Time

Comparisons-Test beds	HTTP		SMTP	
	10 KB	1 MB	10 KB	1 MB
T2 vs. T1 (SC)	5 ± 12%	5 ± 8%	2 ± 5%	3 ± 18%
T2 vs. T1 (FC)	8 ± 7%	5 ± 1%	9 ± 5%	5 ± 5%
T2 vs. HC	1 ± 6%	1 ± 0%	2 ± 5%	1 ± 1%

Results for scenarios using test bed configuration 1 with the slow client had the greatest deviation from results for test bed configuration 2 and results for the heterogeneous test bed had the least deviation. This was because the heterogeneous test bed had the same characteristics as the wireless test bed configuration 2, as described above. The results also showed that if the deviation from the mean value is small, measuring one value in a different network topology could enable the

prediction of the increase in transfer time due to the IPSec overhead. This deviation would be dependent on the processing capabilities of the nodes involved. In our test bed, the server had higher processing capabilities than the client. The client could not process received data fast enough and sent ACKs to slow the flow of packets. As a result, the deviation was greater between different IPSec scenarios. However, in other networks, especially a wireless network or other network with limited link capacity, it would be more likely that the nodes will have time to process the data received and the deviation will tend to be less.

4.3 Overhead of ESP versus AH with 3DES

Switching from ESP to AH authentication (using 3DES) did not significantly affect the number of transactions. The percentage increase in the network load affected mostly the 1 KB file by approximately 6% and, to a lesser extent, the larger files. The transfer time was obtained by averaging the sum of the percentage increase in transfer time for two comparisons for each file size. The maximum percentage increase of transfer time was approximately 3%, which was relatively insignificant.

4.4 HMAC-MD5 versus HMAC-SHA1

The percentage increase in the network load and the number of transactions for MD5 and SHA1 was the same regardless of the authentication algorithm used. However, sending the files using SHA1 always required a longer transfer time due to its higher computational requirements. The values obtained also depended on the computational capabilities of the computer used. For test bed configuration 1, the ranges for the relative increase in transfer time were 1% to 4% for the fast client, and 3% to 9% for the slow client.

5. Conclusions

This research provides a comprehensive compilation of IPSec overhead data with regards to various scenarios, protocols, algorithms, media, and file sizes.

HMAC-SHA1 required a maximum of a 9% increase in transfer time when compared to HMAC-MD5. Even though this percentage increase may not be as significant on a wireline network, it is still of marked importance as it may have a greater impact on environments with low processing capabilities and limited bandwidth that required faster transfer times.

In addition, AH authentication and ESP encryption had a higher percentage increase in network load for the small files only when compared to ESP encryption and ESP authentication. Therefore, the choice of using ESP for authentication and encryption versus AH for authentication and ESP for encryption depends on the frequency of smaller versus larger files that are transferred and on the bandwidth of the links used in a network. That is, if the percentage of small files sent over a network is significant and the network has limited bandwidth, then it would be better to use ESP instead of AH to provide authentication. However, this decision also depends on the security policy of the network and whether it is allowed to forgo the extra authentication coverage provided by AH.

Finally, when IPSec was used over a wireless medium, the network load and number of transactions were the same as in a wireline environment. The transfer time increased, since more time was required to send the additional overhead due to IPSec over the wireless link. It was also shown that it is possible to predict the increase in the transfer time due to IPSec by comparing the transfer time for different network topologies, as long as the number of transactions and the network load used for a specific encryption scenario remain approximately the same.

This research did not consider the processing overhead for compressing files before sending them. Newer FreeS/WAN IPSec implementations can be used to measure the impact on the network overhead when both compression and security are used [5].

6. Acknowledgment

This research was supported in part by the Office of Naval Research through the Navy Collaborative Integrated Information Technology Initiative (NAVCIITI).

7. References

[1] J.P. McGregor and R.B. Lee, "Performance Impact of Data Compression on Virtual Private Network Transactions," *Proc. 25th IEEE Conf. Local Computer Networks*, 2000, pp. 500-510.

[2] B. Chappell, D. Marlow, P. Irey, and K. O' Donoghue, "An Approach for Measuring IP Security Performance," *IPPS/SPDP Workshops – Parallel and Distributed Processing*, 1999, pp. 389-394.

[3] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol," RFC 2401, The Internet Society, Nov. 1998.

[4] M. S. Johns, "Identification Protocol," RFC 1413, Internet Engineering Task Force, Feb. 1993.

[5] "Linux FreeS/WAN," http://www.freeswan.org/ (current May, 2002).