A Performance Study of IPSec Protocol

S. A. Shaikh

Broadlands 202, Park Campus,
Department of Multimedia & Computing,
University of Gloucestershire Business School,
Cheltenham Spa, GL52 2QF, UK
Tel: +44 (0)124 254 3354
Fax: +44 (0)124 254 3327
Email: sshaikh@glos.ac.uk

S. Al-Khayatt

School of Computing & Management Sciences
Sheffield Hallam University
City - Campus, Howard St,
Sheffield, S1 2WB, UK
Tel: +44 (0)114 225 3768
Fax: +44 (0)114 225 3161
Email: s.alkhayatt@shu.ac.uk

Abstract

Internet Protocol Security (IPSec) is a protocol suite that provides a secure way of communicating over the TCP/IP protocol. The IPSec protocol is a set of security extensions developed by the IETF providing what is known as "packet-level security". It provides an extensive set of configurations and tunnelling techniques. The choice, however, of authentication and encryption techniques and configurations affects issues such as data throughput and performance of the network. This paper investigates the performance of IPSec tunnelling. An experimental network is used as a test setup to analyse the performance of IPSec, looking at various combination of authentication and encryption algorithms, hardware offloading of cryptographic processing and the impact of streaming multimedia traffic.

Keywords: IPSec, security protocol, performance

1 Introduction

The Corporate Information Systems Department (CIS) [1] at Sheffield Hallam University (SHU) [2] is responsible for maintaining the networking infrastructure on the university's campuses providing computer information system services to the users at the university. CIS maintains dedicated segments of the network for administrative departments of the university. Although staff members can access them from specified machines by CIS, users require more flexible access. An ideal situation for users would be that they may work securely from any workstation within SHU. This may be extended to external machines outside SHU.

In an attempt to investigate and evaluate a few possible solutions, the Department of Computing and Management Sciences (CMS) carried out a study in collaboration with CIS. The study aimed at evaluating IPSec [3] as a protocol that provides selective end-point-to-end-point encryption on public networks. The implementation and performance aspects of IPSec, in large networks with wide ranging access to thousands of users, were of particular concern to CIS.

IPSec [3] is a suite of protocols providing a set of IP extensions for implementing security at the network (IP) layer. This is important as IP networks have provided the foundation for the modern worldwide networks, and the "network of the networks", the Internet. It offers effective key negotiation and exchange, which makes it very suitable for scalable deployment and public key infrastructure (PKI).

The rest of this paper is organised as follows. Section 2 describes some of the basic components of IPSec in detail relevant to our discussion. Section 3 describes the experimental network and the test setup used for the performance tests. Section 4 analyses the results. Section 5 finally concludes the paper.

2 Overview of IPSec

The purpose of this section is to explain the underlying concepts of IPSec. It combines two main protocols Authentication Header (AH) [4] and Encapsulating Security Payload (ESP) [5]. Both of them play a different role and although they work together, they are independent protocols performing independent functions. We discuss the functionality of each and how they coordinate with each other to provide secured communications over IP.

2.1 Operational modes of IPSec

IPSec operates in two modes: Transport and Tunnel mode. The difference lies in the way the IP packets are processed and how the IPSec Security Associations (SA) are formed between the communicating machines. In transport mode, only the IP payload in an IP packet is encrypted. The extra bytes added to the original packet are less than the ones added in the tunnel mode in which the entire original IP datagram is encrypted and becomes the payload in a new IP packet as shown in Figure 1 below.

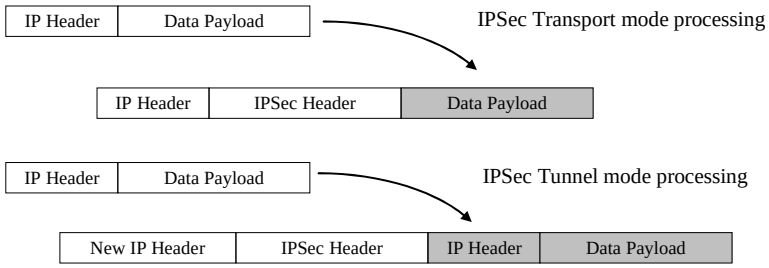

Figure 1: IPSec packet processing in two IPSec modes

The transport mode allows the devices on the public network to see the final source and destination of the IPSec packet. This allows intermediate networks to enable any special features such as Quality of Service (QoS) for Voice over IP (VoIP) services; any processing on the Transport Layer cannot be performed as everything above the IP layer is encrypted. In case of tunnel mode, some networking services can be introduced for special purposes. Any network device located on the perimeter of the network can act as a stand-in IPSec processing peer, allowing the machines behind the device to be spared from any IPSec processing. Such a device can also run a Network Address Translation (NAT) service hiding the IP addresses behind it. One of the advantages of IPSec operating in tunnel mode is that if the routers or firewalls on the edge of the networks are performing the IPSec processing then the end systems do not need to be modified in anyway, which is particularly useful for large deployments of IPSec.

Top

2.2 Authentication Header (AH)

The AH protocol [4] ensures that every packet is authenticated and maintains integrity. It applies cryptographic hash to the packet data and identification information (source/destination address), and attaches it to the packet. The AH header comes after the basic IP header as shown in Figure 2 below. The hash is applied to the shaded parts of the packet. The implementation differs from the transport to the tunnel mode for reasons explained earlier.

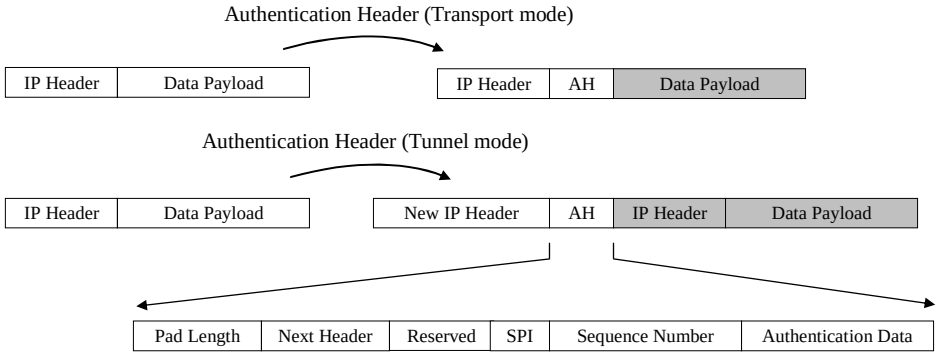

Figure 2: Packet processing involved in Authentication Header

There are six main fields in the AH Header. The *next header* indicates what higher level protocol follows the AH. The *payload* specifies the size of the AH Header in an 8-bit field. The *reserved* field is currently left blank but is reserved for any future use. The *security parameter index* (SPI) notifies the receiving IPSec peer which SA must use to handle the packet. The *sequence number* provides a count for every packet received (incremented for each packet sent) and, therefore, provides anti-replay security. The *authentication data* field is the actual digital signature for the packet, which includes padding to settle the packet size if required.

2.3 Encapsulating Security Payload (ESP)

The ESP protocol [5] reconstructs the IP payload in an encrypted form. The ESP header does not consider the fields of the IP header preceding it and therefore only guarantees the security of the payload. An ESP header may also provide authentication for the payload (in addition to AH). The packet processing for ESP is exactly the same as AH. AH and ESP may not be applied together but if they are then the ESP header follows the AH Header allowing AH to perform the hash function on the entire packet including the ESP header.

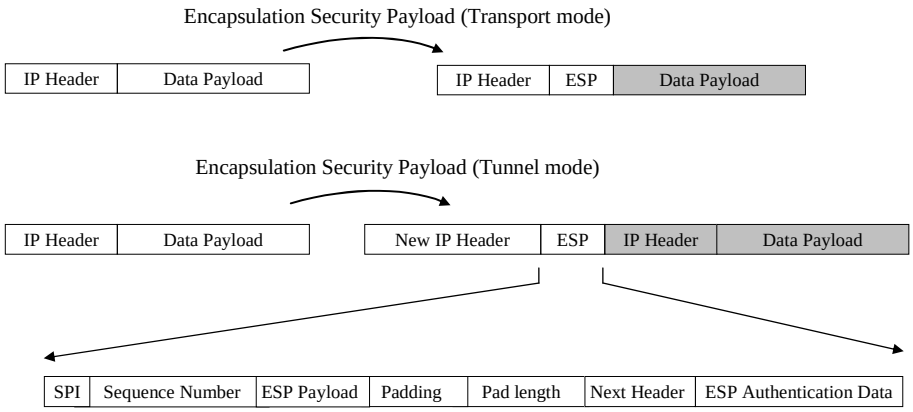

Figure 3: Packet processing involved in Encapsulating Security Payload

Top

Figure 3 above shows the ESP packet processing. The ESP header has six fields. The *SPI* notifies the receiving IPSec peer which SA is to be used to handle the packet. The *sequence number* provides a count for every packet received and is increased for each packet sent, which provides anti-replay security. The *ESP payload* is the actual encrypted payload being carried by the packet. The *padding* field ranges from 0 to 255 bytes of data allowing certain types of encryption algorithms to process the data, which is required to be a multiple of a certain number of bytes. The *pad length* specifies how much of the payload is padding as opposed to data. The *next header* field indicates what higher-level protocol follows the ESP. The *ESP authentication data* field is part of the authentication data in case ESP is configured to provide authentication.

3 Performance Test Setup

This section describes the test network that was used for our experiments. Figure 4 shows the physical setup of the network. The network is divided into three subnets. Each of the subnets is different with respect to the security of the data that is going through them and the users and devices that are the part of the subnet.

The setup in Figure 4 depicts an orthodox Firewall/VPN configuration. A DMZ (De-MilitariZed) subnet is positioned between two subnets, the public and the private subnet. It hosts a router-based-firewall on a Cisco 2621 router [6]. It also hosts a Windows 2000 Server [7] acting as a VPN gateway. The purpose of this is to provide a secure communication tunnel with the users who are placed on the public subnet. This provides a means of accessing the resources on the private subnet.

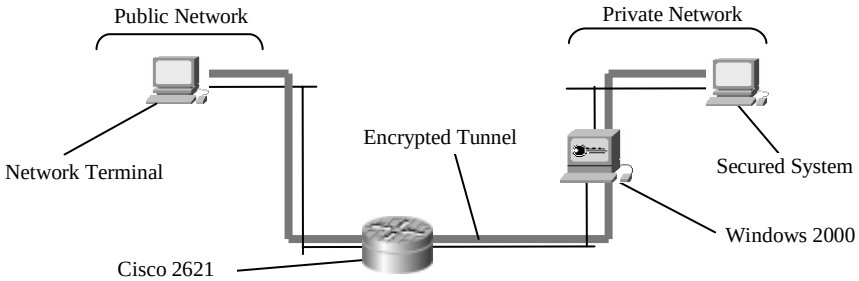

Figure 4: Test setup

The purpose of conducting such tests is to evaluate the IPSec Tunnels and observe how different configurations play a role in their performance. An attempt will be made to determine performance levels for the test network setup with and without encryption. The throughput determined in kilobytes per second (kbps) reflects the efficiency of the communication channel. The throughputs calculated for different configurations are then compared with each other. The higher the throughput is the more effective and quicker the configuration. Differences in throughputs achieved are also calculated in percentages for the purpose of comparison.

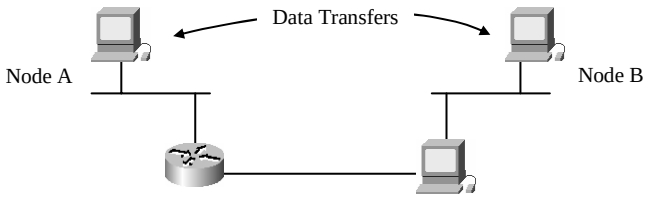

Figure 5: IPSec packet processing in both modes

Top

To determine the performance of the network, two nodes were selected: node A and B (refer to Figure 5). Data was transmitted from node A to node B. To calculate the throughput, files were transferred using File Transfer Protocol (FTP) with time recorded to transfer the complete file. Three different file sizes are transferred, that is, 100, 200 and 400 MB. For each file size, the file was transferred three times with the average time of the three transfers being recorded. The data throughput is calculated. Windows 2000 IPSec implementation involves the two IPSec peers, the two nodes A and B. The role of the intermediate node - the router - remained the same throughout the tests acting as a packet forwarding router in both directions.

4 Performance Evaluation

This section discusses the different tests conducted along with the results obtained from those tests.

4.1 Latency in IPSec Tunnels

One of the first assessments of IPSec investigates the latency introduced by the IPSec tunnels into the performance of an ordinary network. The results are given in Table 1 below.

Table 1: IPSec Tunnel Latency

IPSec Tunnel Latency	
	Data Throughput (Kbps)
Without Tunnelling	3317.16
With Tunnelling	3154.88
Latency	4.89
Loss in Performance	4.89%

The tunnelling latency introduced during the tests is only 4.89%. The data was transferred between the two nodes, originated from node A and travelled in the direction of node B. The node A was running a FTP Service. The CPU utilization on the node A was also observed, ranging between 4 – 26% when no tunnel was applied and between 14 – 62% once the tunnel was in operation. This is because of the IPSec processing load that the node carries. The channel utilization was also affected but only slightly. It ranged between 3.4 - 13.8% compared to 4.9 – 14.9 % when no tunnelling was applied.

4.2 Hardware-supported processing

We compare the above set of results to when IPSec processing is offloaded onto hardware, that is, special Network Interface Cards (NICs). We observe some surprising results shown in Table 2 below.

Table 2: IPSec Tunnel Latency with hardware support

IPSec Tunnel Latency (with hardware-supported processing)	
	Data Throughput (Kbps)
Without Tunnelling	3317.16
With Tunnelling	3568.83
Latency	(-)7.58
Gain in Performance	7.58%

The increase in performance is surprising and unexpected. Offloading most of the IPSec processing on these special NICs was expected to bring about improvement in the performance of the tunnelling but to an extent that it surpasses the unencrypted throughput proved to be a surprise. The throughput achieved by using dedicated hardware supersedes the performance of the network even without any tunnels. This asserts the importance of using any low-level hardware processing, that is considered to be much faster than its high-level counterpart. Moreover, when in hardware-offload mode, the NICs also offloads the TCP and IP checksums. We discuss that in detail in [8].

The channel utilisation is not significantly affected by the use of IPSec-offload. The range observed is 3.35 – 14.8% when the tunnel is applied, not very different without the tunnel which ranged between 4.92 – 14.98%. These results emphasise the fact that the IPSec processing load on the CPU during IPSec tunnelling affects the performance hugely. The CPU load goes as high as 62%. This may be one of the reasons why hardware offloading of IPSec achieves increased throughput. This claim however is rather weakened, when the CPU utilizations observed during this test are considered, which ranged between 22 - 68%. The most likely conclusion that can be drawn from this is that these special IPSec-offloading NICs are efficient in achieving better results but work very much along with the CPU. They certainly raise the network throughput but how much contribution is made towards this by IPSec offloading is not clear.

4.3 Algorithm-dependent performance

The choice of algorithms provided cannot be considered very generous by Windows 2000 as it only provided 64-bit DES and 192-bit Triple DES [9] for encryption. The authentication methods provided are SHA-1 160-bit [10] and MD5 128-bit [11]. The performance of the IPSec tunnels using a varying combination of these algorithms was tested. Table 3 below shows the throughputs obtained.

Table 3: IPSec Tunnel Performance with algorithm combinations

IPSec Tunnel Performance			
Encryption	**Authentication**	**Throughput (Kbps)**	**Latency %**
DES 64-bit	SHA-1 160-bit	3154.88	4.89
DES 64-bit	MD5 128-bit	2635.37	20.55
Triple DES 192-bit	SHA-1 160-bit	2348.47	29.20
Triple DES 192-bit	MD5 128-bit	2543.63	23.32

The triple DES was slower than expected. Triple DES may not involve exactly three times the processing as DES involves, but it does affect the performance of the tunnel considerably.

Top

The varying combination of the encryption and authentication algorithms does reveal an interesting set of results. MD5 when combined with DES is slower than SHA-1 in contrast to that when MD5 is combined with triple DES, it is apparently faster. The relation may be supported by the results obtained with the hardware-supported IPSec tunnels shown in Table 4. Here the pattern is repeated.

Table 4: IPSec Tunnel Performance with algorithm combinations and hardware support

IPSec Tunnel Performance (hardware offload)		
Encryption	**Authentication**	**Throughput (Kbps)**
DES 64-bit	SHA-1 160-bit	3568.83
DES 64-bit	MD5 128-bit	2978.04
Triple DES 192-bit	SHA-1 160-bit	2548.60
Triple DES 192-bit	MD5 128-bit	2668.04

The message digest produced by SHA-1 is 32-bits longer than MD5; hence SHA-1 is considerably stronger than MD5. MD5 is generally considered to be slightly more vulnerable whereas SHA-1 is generally believed to be resistant to cryptanalysis. One reason is that the design criteria of SHA-1 are not public so the security is difficult to judge. MD5 has 4 rounds of 16 steps and a bit-length of 128, while SHA-1 has 4 rounds of 20 steps and a bit-length of 160. SHA-1 compared to MD5 will execute slightly slower as it has more steps and a longer buffer.

When combined with triple DES, SHA-1 complies with the theoretical basis of its execution and is slower than MD5 but with triple DES the results suggest otherwise. One of the reasonable explanations can be the fact that due to it having a larger buffer, some extra data may be processed in each payload unit, providing quicker processing of data. The IPSec-offloading does affect the performance for single DES but not much difference is noted between the throughputs obtained with triple DES. The extra amount of processing does not allow the NICs to provide any better efficiency. The comparison between the hardware offload and CPU-based processing of different algorithm combinations is shown in Table 5 and in Figure 6.

Table 5: A comparison of IPSec processing: CPU-based and hardware offload

IPSec Processing				
Encryption	**Authentication**	**CPU-based**	**Hardware offload**	**Improvement %**
DES 64-bit	SHA-1 160-bit	3154.88	3568.83	13.12
DES 64-bit	MD5 128-bit	2635.37	2978.04	13.00
Triple DES 192-bit	SHA-1 160-bit	2348.47	2548.60	8.52
Triple DES 192-bit	MD5 128-bit	2543.63	2668.04	4.89
Average improvement (due to hardware offload)				9.88

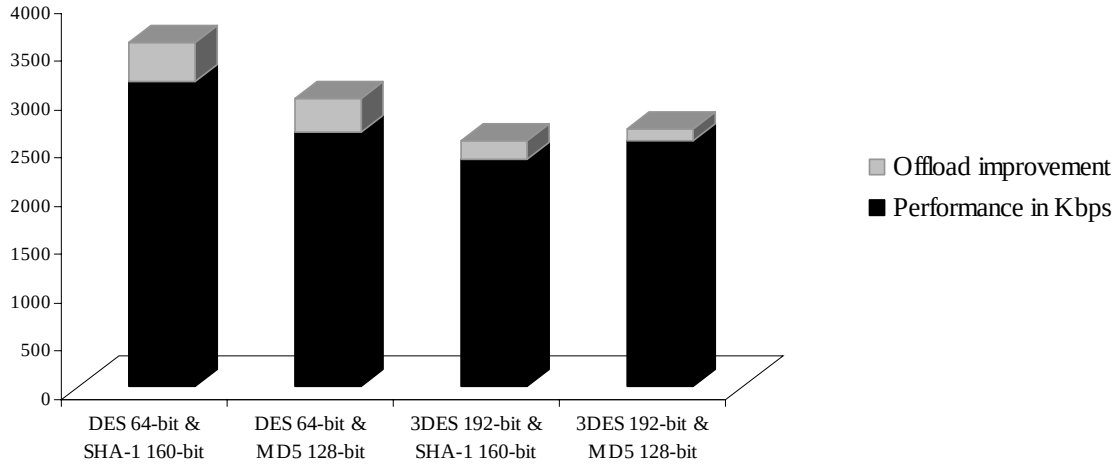

Figure 6: IPSec processing with hardware offload improvement

An average improvement of more than 9% stresses the efficiency of the hardware offloading of IPSec processing. The load that the CPUs have to bear in order to sustain IPSec tunnels has already been observed earlier. Any offloading of these functions improves the performance considerably.

4.4 Varying key-update intervals

The IPSec SAs have specified lifetimes after the expiration of which a new SA is renegotiated. The SAs can be configured to expire according to whichever comes sooner either the number of specified seconds passed or the amount of traffic (specified in kilobytes) passed. These standards might be different in every implementation of IPSec, but most of the popular IPSec implementation including that of Microsoft [7] and Cisco Systems [6] conform to it.

The main purpose of this is to update cryptographic keys for each session. When an IPSec peer receives a SA negotiation request from the peer, it will use the smaller of either the lifetime values proposed by the requesting peer or the locally configured lifetime value for new security associations. The renegotiation of these SAs and the key updates has an effect on the performance of the tunnel as some extra traffic is generated. Tests were performed to investigate this and estimate the magnitude of the effect.

During secure communications, a device may form many SAs, each SA being formed between only two devices. In order to communicate with many devices, many SAs will be formed and therefore a mechanism is required to keep track of these SAs. A security parameter index (SPI) identifies each SA uniquely. This is a 32-bit random number that the system allocates to every SA to represent it during SA negotiations. It is tangible compared to a SA which is only in concept. The lifecycle of a SPI works out to be such that when a device ‘A’ starts SA negotiation with another device ‘B’ the receiving device ‘B’ will assign a SPI to the SA being formed. The device ‘B’ then forwards that SPI to the initiating device ‘A’. From there onwards, until the end of the SA lifetime, whenever device ‘B’ wishes to communicate with the device ‘A’ using that SA, it uses that SPI to specify it. Device ‘A’, on receiving the SPI determines which SA it needs to use and processes the packets according to the terms of

the SA. Due to the inter-communication of SPI, the SPI is not encrypted in the packet. The results for the test are shown below in Table 6.

Table 6: Performance of various key-update interval settings

Varying key-update intervals for IPSec tunnels	
Kilobytes	**Throughput (Kbps)**
50000	3010.7
100000	3028.5
200000	3056.8

The results show an increase in the data throughput as the interval doubles every time. As noted earlier this is due to the reduction in processing delay caused by the renegotiation of encryption keys. The measurements taken were only for three intervals. The main reason for this is because the sample file that was used to transfer what across the network was only 400 MB in size. The largest interval being used is 200 MB. This means that if this interval is doubled or increased any further then the encryption keys will only be negotiated twice. If the interval is 400 MB, then the negotiation will only take place once. It is interesting to see that as the key-update intervals increase the performance of the network also improves; the communication overhead decreases as the key-update intervals grow larger.

4.5 Streaming Multimedia over IPSec

The real objective behind testing the tunnels for multimedia is to reveal the affects on performance by varying the network traffic and services. It is important to observe the differences on the tunnel performance by real-time multimedia streaming as it creates major network-burdening traffic and therefore is a cause for concern in large public networks.

Referring to the Figure 5, a Real Server [13] is used at node A providing a live streaming RealVideo to a RealVideo client [13] on the node B. The streaming video was initially played without encryption and then played through an encrypted tunnel. The Real Server provided detailed statistics about the streaming connections that are established for transmitting multimedia. The Real Server administrator provides a web interface for its configuration and monitoring. It determines statistics such as the CPU usage, memory usage, bandwidth usage, players (clients) connected and the files being currently used. In order to monitor the actual CPU usage, the “CPU Usage History” provided by the Windows 2000 operating system was used.

The test conducted involved transmitting a single streaming connection through an encrypted tunnel. It was compared to similar streaming without encryption. The file used was 4286 KB in size and provided a RealVideo stream of 2 minutes and 37 seconds duration. The bandwidth was constant at 220 Kbps throughout the connection of the Real player, both with and without encryption. The algorithms used were DES (64-bit) and SHA-1 (160-bit), a standard, for the tunnel. The encryption key renegotiation interval was kept at the maximum at 65535 packets, which was far larger than the number of packets that travelled the network through out the test. This ensured that no extra processing delays were introduced. It was also ensured that the Real player did not perform any caching of multimedia files. This was done in order to make sure that file was requested from the server every time it was played. The results obtained can be seen in the Table 7 below.

Table 7: Performance of streaming multimedia

Streaming multimedia performance				
Unencrypted Multimedia Traffic	Lowest	Highest	Average	
Throughput achieved	211.9	792.2	355.9	Kbps
Channel Utilization range	0.10 - 0.34 %			
IPSec Peer Utilization	0 - 2 %			
Encrypted Multimedia Traffic	Lowest	Highest	Average	
Throughput achieved	206.6	621.5	354.8	Kbps
Channel Utilization range	0.11 - 0.38 %			
IPSec Peer Utilization	0 - 4 %			

The average throughput obtained without encryption is more than the one achieved using the encrypted tunnel. One explanation is that when the data packets are encrypted, they are bigger in size. So the throughput obtained by the encrypted multimedia traffic, reflects the extra burden carried by the network. The encrypted tunnel also causes the channel utilization to increase as more traffic is transmitted through. The IPSec peers also show a slight increase in the CPU utilization.

5 Conclusion

This paper has attempted to analyse the network performance for the IPSec protocol. We have looked at various aspects of deploying IPSec, including various configurations of authentication and encryption algorithms, varying key-update intervals and multimedia traffic. Most interesting, however, have been results obtained for hardware offloading of cryptographic processing as it raises some interesting issues.

Performance testing secured communications and networks requires consideration and care in order for the results to be accurate and worthy. The test setups used in this study were inspired by similar attempts made in [14] and [15]. It is important to consider more controlled and accurate test setups and adopt more precise testing procedures such as those in [16].

For the purpose of security, it is worthwhile to investigate and evaluate the cryptographic properties of the IPSec protocol; some important weaknesses have been highlighted elsewhere [17]. Whether these weaknesses relate to the security performance in the IPSec standard is of interest. More attention may also be given to supporting infrastructure when it comes to securing applications with IPSec. This paper has considered a wired IP network; it will be interesting to see the performance measures of IPSec tunnels between ubiquitous and cellular devices. Since new wireless standards such as IEEE 802.11b [18] [19] and HomeRF [20] have emerged, it will be interesting to see how IPSec fares with these new standards. Some related work has also been conducted elsewhere [21].

References

[1] Corporate Information Systems (CIS), SHU. At http://www.shu.ac.uk/services/cis, 2004

[2] Sheffield Hallam University (SHU), Sheffield, UK. At http://www.shu.ac.uk, 2004

[3] Kent, S. and Atkinson, R Security Architecture for the Internet Protocol. Internet Request

for Comments RFC 2401, Internet Society, Network Working Group, 1998

[4] Kent, S. and Atkinson, R. IP Authentication Header (AH). Internet Request for Comments RFC 2402, Internet Society, Network Working Group, 1998

[5] Kent, S. and Atkinson, R. IP Encapsulating Payload (ESP). Internet Request for Comments RFC 2406, Internet Society, Network Working Group, 1998

[6] Cisco Systems Inc. At http://www.cisco.com, 2004

[7] Microsoft Windows 2000. At http://www.microsoft.com/windows2000, 2004

[8] S. A. Shaikh and S. Al-Khayatt. Performance of authentication and encryption algorithms in IPSec Tunnelling, Fourth Symposium on Communication Systems, Networks and Digital Signal Processing, CSNDSP'04, Newcastle-upon-Tyne, UK, 2004

[9] Data Encryption Standard (1999) FIPS PUB 46-3, U.S Department of Commerce/National Institute of Standards and Technology. At http://csrc.nist.gov/fips/fips46-3.pdf, 2004

[10] The Use of HMAC-SHA-1-96 within ESP and AH. RFC 2404. At http://www.ietf.org/rfc/rfc2404.txt, 2004

[11] The Use of HMAC-MD5-96 within ESP and AH. RFC 2403. At http://www.ietf.org/rfc/rfc2403.txt, 2004

[12] Harkins, D. and Carrel, D. The Internet Key Exchange (IKE). Cisco Systems Inc., RFC 2409, 1998

[13] Real. At http://www.real.com, 2004

[14] S. Al-Khayatt, S. A. Shaikh, B. Akhgar and J. Siddiqi. A Study of Encrypted, Tunnelling Models in Virtual Private Networks, Third IEEE Conference on Information Technology ITCC'02, Las Vegas, Nevada, USA, April 8-10, 2002

[15] S. Al-Khayatt, S. A. Shaikh, B. Akhgar and J. Siddiqi. Performance of Multimedia Applications with IPSec Tunnelling, Third IEEE Conference on Information Technology ITCC'02, Las Vegas, Nevada, USA, April 8-10, 2002

[16] S. A. Shaikh and S. Al-Khayatt. Analysis and Implementation of Virtual Private Network Support in Corporate Networks, IASTED International Conference on Parallel and Distributed Computing and Networks, PDCN'04, Innsbruck, Austria, Feb 17-19, 2004

[17] N. Ferguson and B. Schneier. A Cryptographic Evaluation of IPSec. Counterpane Internet Security, Inc. At http://www.counterpane.com, 2004

[18] Specification of the IEEE 802.11b Standard. At http:// standards.ieee.org/getieee802, 2003

[19] S. A. Shaikh and S. Al-Khayatt. A Wireless Network for Multimedia Applications – a

Case Study, Third Symposium on Communication Systems, Networks and Digital Signal Processing, CSNDSP'02, Staffordshire, UK, 2002

[20] Specification of the HomeRF System. At http://www.homerf.org, 2003

[21] J. Caldera, D. De-Niz and J. Nakagawa. Performance Analysis of IPSec and IKE For Mobile IP on Wireless Environments. Information Networking Institute, Carnegie Mellon University. At http://www.cs.cmu.edu/~dionisio/ipsec-wmip.pdf, 2003

Top