Performance Comparison of IPsec and TLS Based VPN Technologies

I. Kotuliak*, P. Rybár** and P. Trúchly*

*Faculty of Informatics and Information Technologies, Slovak University of Technology, Bratislava, Slovakia
**Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Bratislava, Slovakia
ivan.kotuliak@stuba.sk, rybar@ktl.elf.stuba.sk, peter.truchly@stuba.sk

***Abstract*—IPsec and TLS based VPN technologies are widely used in nowadays networks. But one can hardly find information about their performance, especially compared to each other. So when there was a speed and delay sensitive interconnection project, a direct performance comparison had to be performed. This article realizes a performance comparison of OpenVPN and IPSec based VPN; we measure what throughput each protocol can provide on given hardware while using the same cipher and key length.**

I. Introduction

The buzzword of this decade in telecommunications is convergence: the convergence of telecommunications, Internet, entertainment, and information technologies for the seamless provisioning of multimedia services across different types of networks. Thus, the future telecommunication network can be envisioned as a group of cooperating heterogeneous fixed and mobile data networks which share a reliable, proven and common Internet Protocol (IP) based backbone. This telecommunication concept based on IP protocol is called IP Multimedia Subsystem (IMS) [1], [2].

The IMS is the unified telecommunication industry approach toward an "All-IP" network architecture that merges the paradigms and technologies of the Internet with the cellular and fixed telecommunication worlds. It aims at creating a reference service delivery platform for provisioning of IP multimedia services in a reliable, secure, and controllable manner. IMS was also adopted as the basis of the Next Generation Networks (NGN) architecture specified by TISPAN [3].

In order to interconnect IMS networks each other and to prevent any security incidents some of VPN (Virtual Private Network) tunneling or encryption should be used. The VPN solutions can be based on e.g. Point-to-Point Tunnelling Protocol (PPTP), IP Security standard (IPsec) or SSL (Secure Sockets Layer) technology [4]. As PPTP solutions are very simple and are also regarded as very insecure, simply because in most implementations there are many not encrypted packets that can be easily spoofed [5] we decided to compare only IPsec and SSL solutions.

Next section contains a brief characterization of SSL and IPsec protocols and their implementations. Section III introduces testbed used for the purpose of comparison and section IV shows results and summarizes coclusions.

II. Protocol Background

A. Brief Protocol Description

SSL is cryptographic protocol that provides secure communication over the Internet [6]. The new version of SSL is called TLS (Transport Layer Security) and it is present in all major web browsers. Its security is provided by using cryptography. TLS is a client/server protocol, its connection starts with a TLS handshake covering negotiation between peers for algorithm support, key exchange and authentication and symmetric encryption and data exchange. TLS encapsulates IP in UDP (User Datagram Protocol). IP packets sent from a virtual network adapter are encrypted and encapsulated onto a UDP connection and sent to a remote host over the Internet. The remote host decrypts, authenticates, and de-encapsulates the IP packets using its virtual adapter.

Like TLS, IPsec is also a set of cryptographic protocols that provide secure communication over the Internet [7]. IPsec connection starts with a two phase handshake and when it is completed an arbitrary traffic can be sent via encrypted tunnel. At start a preliminary secure tunnel is created by using of a handshake protocol called an Internet Key Exchange (IKE) [8]. This IKE process authenticates the end points of the tunnel to each other, and securely exchanges the necessary information to create a more permanent tunnel using symmetric encryption. IPsec has two modes: transport mode, which protects only the transported data, and tunnel mode, which also protects the IP header. Client-to-LAN connections typically use the transport mode, while LAN-to-LAN connections typically use tunnel mode.

B. Protocol implementation

IPsec is very flexible and can be used in many ways. IPsec is used to create a majority of the VPN products found today. Checkpoint VPN-1, Cisco PIX, and the open source OpenSWAN are all examples of commonly used VPN solutions that implement IPsec. However, in addition to configuration complexity, IPsec has not strayed interference with kernel space [9]. This principle breaks out the OS into rings of privilege. Ring0 is reserved for the kernel and other essential processes. Ring1 is reserved for other system processes that require low level access to hardware. When moving outward in rings, the privilege of the process is decreased. Ring3 is where most user processes, including TLS implementations, are found. The rules of architecture state that processes in higher numbered rings cannot interfere with processes in lower numbered rings. This provides greatly enhanced stability and security in our applications and allows for multi-user, multithreaded systems. However, IPsec needs low level access to the interface when it modifies IP headers. It operates in ring0.

OpenVPN is open source SSL VPN implementation for Linux and is the major player in SSL VPN field. There are

978-1-4577-0052-1/11/$26.00 ©2011 IEEE

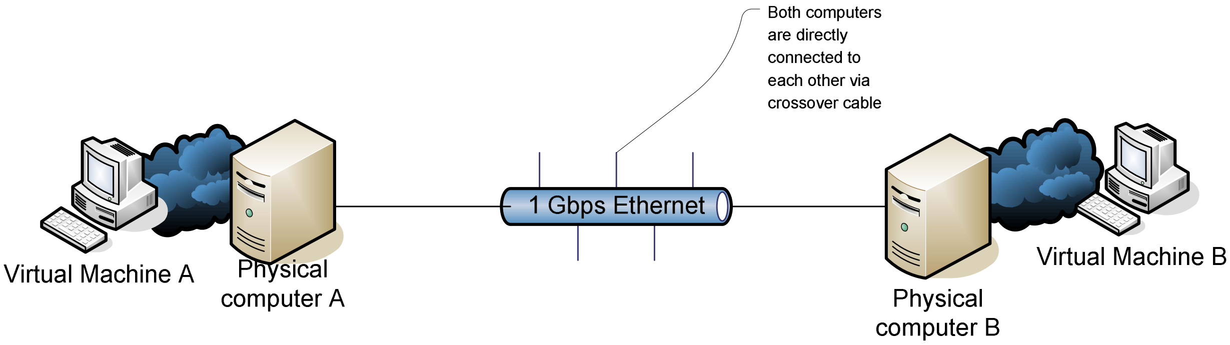

Figure 1. Experimental testbed

other commercial solutions, but none of them are used as widely as OpenVPN and are not always compatible. OpenVPN uses the widespread SSL/TLS protocol to handle tunnel creation and cryptographic elements necessary to create a VPN (the same kind of VPN that IPsec creates) [10]. The main difference is that OpenVPN does not operate as close to kernel as IPsec in user space. OpenVPN does not need to be that close to kernel, because it uses a small "trick". Unlike IPsec, which requires access to the network interface, OpenVPN creates a virtual interface which it can access without kernel dependence and thus it is a little more secure and prone to vulnerabilities by design. The other advantage is that it can be ported more easily to other systems and it runs on Windows, Linux and various Unix versions and Macs.

More computer and VPN users are aware of IPsec than OpenVPN. Therefore we would like to point out that SSL/TLS based VPNs are able to encrypt link traffic in the same way as IPsec VPNs. If looked on the handshake from cryptographical point of view, it uses the same principle of Diffie-Hellman problem as is used by IKE in IPsec. The SSL crypto library is then used to secure the symmetric tunnel, again using similar encryption techniques to those protecting IPsec tunnels [11].

III. Testbed Setup and Performance Parameters

A. *Testbed Setup*

In order to perform performance comparison for aforementioned technologies we decided to realize real-world network tests between two computers running the Linux Debian operating system (Fig. 1). Usually, in a real world network, there are multiple network nodes (more than two). This, however, is not necessary in our case. We were trying to find out what the computational complexity (and thus throughput) of these two encryption methods was. We also reviewed various network parameters of the implementation of an encrypted tunnel, such as how encryption affects response time.

The performance comparison was made with help of a program called IxChariot. IxChariot is widely used to test network equipment under various traffic patterns. It consists of a console which manages so called endpoints. The endpoints generate (or receive) traffic and report the results to the console [12].

The tests were done using two identical Linux Debian systems running as Virtual machines under the Vmware virtualization program. Windows Vista was used as a host operating system on two physical PCs. Although this does not usually perform very efficiently, it did serve our purpose. The extra computational overhead caused by virtual machine added up and better showed the difference of both algorithms. The tests use IxChariot script with the file size set to 1,000,000 bytes. The tests were performed in both directions – from Virtual Machine A to B and vice versa.

The use of virtual machines in network testing might seem a strange choice at first because they have the disadvantage of additional processing overhead. One can imagine this overhead as a thick layer of software between our network test procedures and the actual hardware. Since there are 2 operating systems in the way, program execution can be almost more complex compared to normal. However, if we look at this fact from another perspective, it outlines the differences between the two technologies we are looking at. As already mentioned, this test is about comparing two similar technologies and if there is a small difference between those two protocols that is difficult to detect, this setup would emphasize them and thus make them better to detect by our network test.

The advantage of using this virtual machines setup was that we could use identical configurations. The second virtual machine (B) was created by a process called cloning. It created an exact replica of the first machine – other than the name and IP address there were no other differences.

B. *Performance Parameters*

We performed the following IxChariot tests to evaluate next performance parameters:

1. Throughput - Throughput is a measure of how fast data flows through the cable. The test sends data from computer to computer, measures how much time it takes, and calculates the result value in Mbps.
2. Response Time - This test measures all delays that are introduced into a data stream (by link, router), and is essentially what one would measure by using the ping command.

IV. Experimental Results

This section contains test results obtained by using the described test methods. The detailed results are shown only for one direction (computer A to computer B) because of size constraints, however all results (for both directions) will be summarized in table later.

A. *Detailed Results - Throughput*

First test covered simple scenario with no encryption, no VPN (from computer A to computer B). The throughput behavior can be seen in Fig. 2 where the

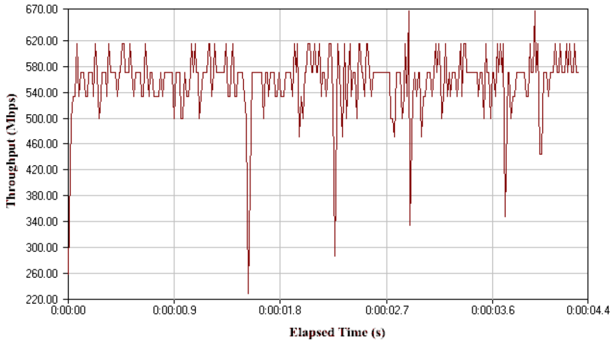

Figure 2. Throughput measured for case with no encryption, A→B

measured average network throughput over the Gigabit network was 553 Mbps. This can be explained by the fact that we did not setup the network to use Jumbo frames. Gigabit Ethernet without the use of Jumbo frames is much lower than its theoretical value.

It is important to mention that the throughput chart shows relatively large variance. It might be due to the nature of Ethernets best effort nature or because of some unknown variables between the physical PC and the virtual machine. However, it is the average value that is important to us. This value can be used as a reference for comparison with the tested setup.

Fig. 3 shows the network throughput when OpenVPN with 3DES cipher was applied. As expected, OpenVPN 3DES exhibits decreased performance. Its average throughput which is approximately 60 Mbps was the lowest recorded for OpenVPN. OpenVPN setup in the test used certificate authority and TLS mode. TLS mode is the most powerful cryptographic mode from a security point of view (and of course that of computational complexity). Comparison with the mode without TLS authentication showed that there was a performance decrease of approximately 5%.

Fig. 4 shows the network throughput when OpenVPN with AES cipher was applied. This test was performed using the same configuration as the previous test, but with a different cipher. We can see that these two ciphers have very similar performance characteristics as for variations. The same can be said of the response time.

The resultant throughput for test with OpenVPN with Blowfish cipher is depicted in Fig. 5. This test was performed using the Blowfish cipher – the default in OpenVPN configuration. We can assume this is going to be a reference configuration for most users. The average value of throughput is around 5.5 times lower than the reference 1 Gbps Ethernet value. The rather large variation in the throughput that we saw in the unencrypted test is here too. We can assume that it is present for the same reasons as in the previous test and is probably not related to OpenVPN. One can say that OpenVPN with Blowfish cipher offers decent throughput at around 96 Mbps.

Most users will probably use the default Blowfish cipher and will not change it to AES, as it does not bring about any notable performance increase or better security. As we have seen, using 3DES is significantly slower and will probably be used only in very specific configurations, or for compatibility reasons.

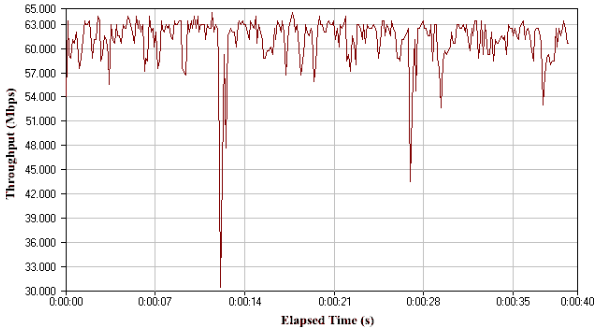

Figure 3. Throughput measured in case with OpenVPN with 3DES cipher, A→B

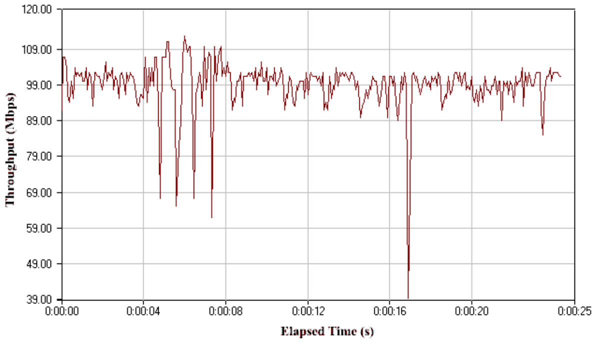

Figure 4. Throughput measured in case with OpenVPN with AES cipher, A→B

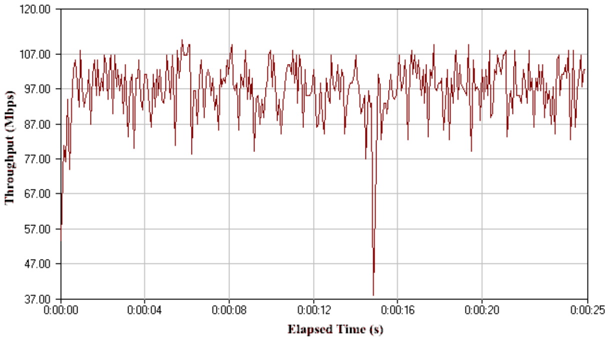

Figure 5. Throughput measured in case with OpenVPN with Blowfish cipher, A→B

Next set of tests was realized with VPN based on IPsec. Fig. 6 and Fig. 7 show charts of the network throughput when 3DES and AES cipher have been utilized. As predicted, IPsec results with AES payload encryption were much better than previous results obtained with default 3DES encryption. AES is clearly superior to 3DES in performance. With the average performance at approximately 140 Mbps, this is a fairly logical result and one that we would have expected. IPsec with its lower level implementation should have algorithms superior to OpenVPN, especially when it comes to speed.

Fig. 8 represents chart of the throughput when Blowfish cipher is applied in IPsec. IPsec with Blowfish cipher showed similar results to those results with AES cipher. The blowfish was unable to overtake AES in throughput, but it shows a lower variance.

B. Overall Results

Now we will summarize and compare consolidated results of all tests performed and say some conclusions

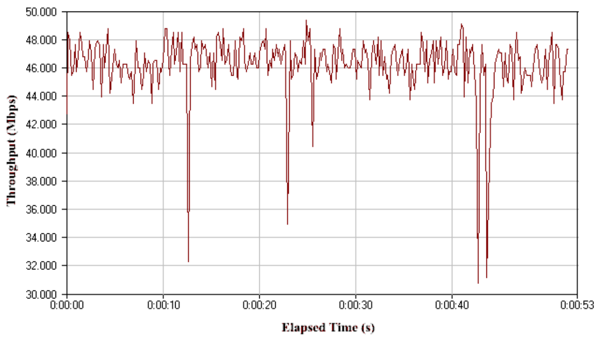

Figure 6. Throughput measured in case with IPsec with 3DES cipher, A→B

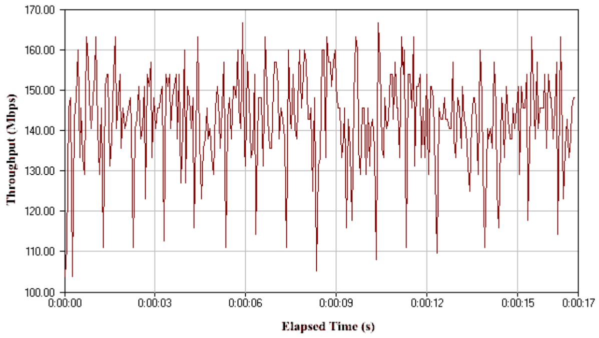

Figure 7. Throughput measured in case with IPsec with AES cipher, A→B

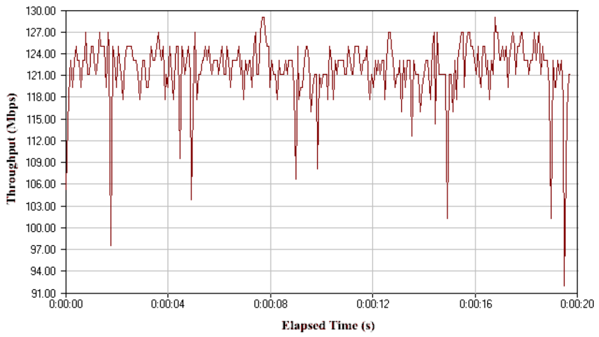

Figure 8. Throughput measured in case with IPsec with Blowfish cipher, A→B

from measured parameters (CPU utilization, throughput, and response time).

The Ixia IxChariot also measured many other link parameters than the one given (e.g. jitter), however there was no significant variation among our tests; therefore these parameters were not further analyzed.

The performance tests we performed show that both OpenVPN and IPsec are capable of creating high performance encrypted links between two or more sites.

As one can see, their performance depends on 3 main factors: the interconnecting link bandwidth, the speed of the encrypting and decrypting device and the type of cipher in use. Our tests show that 3DES is a cipher of the past, and should not be used today. It offers similar (or even less) security to modern ciphers such as AES or Blowfish.

Blowfish and AES offer very similar performance and security. There is no known cryptanalysis and they both can saturate 100 Mbps link on current desktop computers.

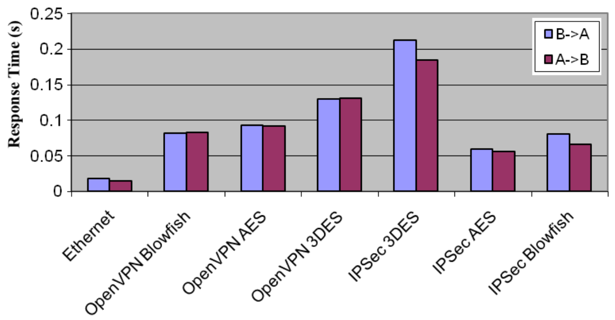

Figure 9. Response time for all realized tests and ciphers for both directions of data transmitting

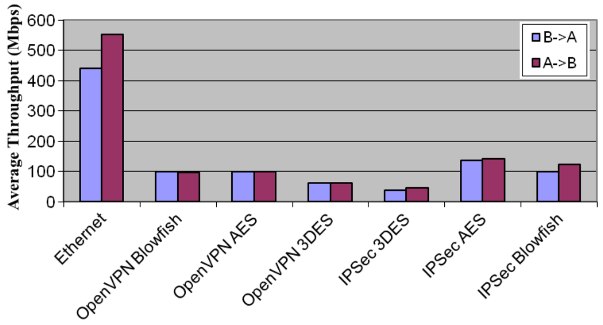

Figure 10. Average throughput for all realized tests and ciphers for both directions of data transmitting

A notable advantage of AES is its standardization and widespread adoption among governments and in the private sector.

Based on Table 1 and Fig. 9 and Fig. 10 we can state that IPsec wins over OpenVPN (while using the same type of cipher) by a rather small margin. It is faster while using AES and Blowfish ciphers and also has smaller delay. It loses only when 3DES cipher is used, but this is not that significant, since this obsolete cipher probably won't be used anymore.

However OpenVPN, and other SSL based solutions, have some strong points too. Most notably, it is ease of use and flexibility. Configuring and installing OpenVPN is a child's play compared to IPsec. Let us take the documentation as an example. IPsec documentation is spread over 6 or more different manual pages, which makes it quite difficult to use. OpenVPN has one documentation file and one "How-to", both of them are on OpenVPN website and also in Linux manual pages. Another area where OpenVPN wins in our opinion is complexity. This point has already been mentioned in this paper, when describing community reactions to the IPsec. We can therefore ascertain that this is true.

V. Conclusion

This paper concentrated on VPN technologies which utilize SSL/TLS or IPsec protocols to create secure tunnel for data transmission, e.g. to interconnect two IMS networks. A several tests have been performed to compare these technologies based on parameters such as throughput, response time and so on. We can summarize that it is difficult to choose the better of these two technologies based on all views. Each user has different needs. For our implementation we decided to choose OpenVPN, due to its simplicity and fast and straightforward implementation. On the other hand IPsec

TABLE I.
Type Sizes for Camera-Ready Papers

Test	Response average (s)	Response maximum (s)	Average throughput (Mbps)	CPU utilization sending node	CPU utilization receiving node
A->B Ethernet	0,014	0,035	553	71	81
B->A Ethernet	0,018	0,038	440	90	59
A->B OpenVPN Blowfish	0,083	0,2	96	66	90
B->A OpenVPN Blowfish	0,081	0,3	99	95	77
A->B OpenVPN AES	0,092	0,19	98	62	94
B->A OpenVPN AES	0,093	0,3	99	93	78
A->B OpenVPN 3DES	0,131	0,263	60,98	78	94
B->A OpenVPN 3DES	0,129	0,352	61,77	92	86
A->B IPsec 3DES	0,184	0,28	45	99	40
B->A IPsec 3DES	0,212	0,37	37,7	99	32
A->B IPsec AES	0.056	0.077	142	90	84
B->A IPsec AES	0.059	0.1	135	97	63
A->B IPsec Blowfish	0,066	0,087	121,76	98	73
B->A IPsec Blowfish	0,08	0,197	99,87	99	52

is somewhat faster and as it has been on the market much longer than SSL VPN solutions and it has far more support among hardware and software vendors.

Acknowledgment

This paper was supported by Slovak national projects VEGA 1/0243/10 and VEGA 1/0720/09.

References

[1] 3GPP, *Service requirements for the Internet Protocol (IP) multimedia core network subsystem (IMS); Stage 1*, 3GPP TS 22.228, 2011, http://www.3gpp.org/article/ims.

[2] 3GPP, *IP Multimedia Subsystem (IMS); Stage 2*, 3GPP TS 23.228, 2011, http://www.3gpp.org/article/ims.

[3] C. Esteve Rothenberg, *Fixed-mobile Convergence in TISPAN/3GPP IMS: Conception and Evaluation of Systems for Seamless Vertical Handover*, VDM Verlag, 2008.

[4] M. Lewis, *Comparing, Designing, and Deploying VPNs*, Cisco Press, 2006, 1080 p.

[5] B. Schneier and Mudge, "Cryptanalysis of Microsoft's Point-to-Point Tunneling Protocol (PPTP)," Proceedings of the 5th ACM Conference on Communications and Computer Security, ACM Press, pp. 132-141.

[6] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", IETF RFC 5246, 2008, http://tools.ietf.org/html/rfc5246.

[7] S. Kent and K. Seo, "Security Architecture for the Internet Protocol", IETF RFC 4301, 2005, http://tools.ietf.org/html/rfc4301.

[8] D. Harkins and D. Carrel, "The Internet Key Exchange (IKE)", IETF RFC 2409, 1998, http://tools.ietf.org/html/rfc2409.

[9] B. Schneier and N. Fergusson, "A crypthographis evaluation of IPsec", 2003, http://www.schneier.com/paper-ipsec.pdf.

[10] M. Feilner, *OpenVPN: Building and Integrating Virtual Private Networks*, PACKT, 2006, 258 p.

[11] CH. Hossner, "OpenVPN and the SSL VPN Revolution", paper from SANS Institute, 2004.

[12] Ixia, "IPsec Virtual Private Networks: Conformance and Performance Testing - Sample Test Plans", 2004, http://www.syrus.ru/files/pdf/ixia/ipsec.pdf.