IPSec Hardware Resource Requirements Evaluation

Alberto Ferrante, Vincenzo Piuri
Department of Information Technologies,
University of Milan
Milano, Italy
Email: {ferrante, piuri}@dti.unimi.it

Jeff Owen
ST Microelectronics Inc.
San Jose, California
Email: jefferson.owen@st.com

Abstract— **IPSec is a suite of protocols that adds security to communications at the IP level. This suite of protocols is becoming more and more important as it is included as mandatory security mechanism in IPv6. In this paper we provide an evaluation of the hardware resources needed for supporting virtual private networking through IPSec. The target system of this study is a home secure gateway, therefore only the tunnel mode is considered. Focus is on ESP protocol, but also some evaluations on AH are provided. We discuss usage of the AES, HMAC-SHA-1, and HMAC-SHA-2 cryptographic algorithms.**

In this paper we show that enabling IPSec in a 100Mbit/s network kills its performance in almost every case. In a 10Mbit/s network the results obtained for performance and CPU usage are much better. An interesting case within this network configuration is that in which IPComp is enabled and used on compressible data: CPU usage grows to 100%, but network throughput rises over the 10Mbit/s limit, due to data compression.

This performance evaluation leads the conclusion that while a hardware crypto-accelerator is really key in reaching high performance, it may also be useful in small, slow systems (e.g. small embedded systems) where it would help improving performance and security.

I. Introduction

IPSec is mainly composed of two protocols, Authentication Header (AH) and Encapsulating Security Payload (ESP). The former allows authentication of each IP datagram's headers or – depending on the operational mode that has been selected – of the entire IP datagram. The latter allows encryption – and optionally authentication – of the entire IP datagram or of the IP payload, depending on the operational mode that has been selected, namely the transport and the tunnel modes. The former was designed for being used in host machines, while the latter is for secure gateways. In tunnel mode the entire original IP datagram is processed; the result becoming the data payload of a new IP datagram with a new IP header. In transport mode only parts of the original IP datagram are processed (e.g. the data payload for the ESP protocol) and the original IP header is kept with some small modifications. Through encryption, authentication, and other security mechanisms included in IPSec (e.g. anti-reply), data confidentiality, data authentication, and peer's identity authentication can be provided [1], [2], [3], [4].

IPSec is often used to create Virtual Private Networks (VPN). A VPN is an extension of a private network on a public network (e.g. the Internet) [5], [6]. The extended part of the network logically behaves like a private one. Typical usage scenarios for VPNs are: remote user access to a private LAN over the Internet and connection of two private networks. In these cases a virtual secure channel needs to be created, respectively, from the user's PC to the LAN public access point or from one LAN to the other. See Figure 1 for the former scenario and Figure 2 for the latter one. Private networks public access points are called *secure gateway*. A secure gateway is a router or a router/firewall also running a VPN-enabled software (e.g., an IPSec implementation and a VPN server). All the traffic inside the LAN is usually not protected, while the traffic going out or coming in the LAN through the secure gateway is protected by some security mechanisms.

Developing a performance and CPU resource evaluation methodology is necessary to understand the possible benefits of using hardware accelerators and, most of all, to evaluate different hardware/software architectures from the performance stand point. Measuring the CPU usage is really important because it allows evaluation of different configurations even when they provide similar throughput.

Very few evaluations of the resource requirements of IPSec have been done so far. In [7] some results about a 1Gbit/s network are reported. The results shown there are about network performance and few indirect evaluations on CPU usage. In [8] some network performance results are reported with respect to an IPv6 network used for

0-7803-8900-X/05/$20.00 © 2005 IEEE

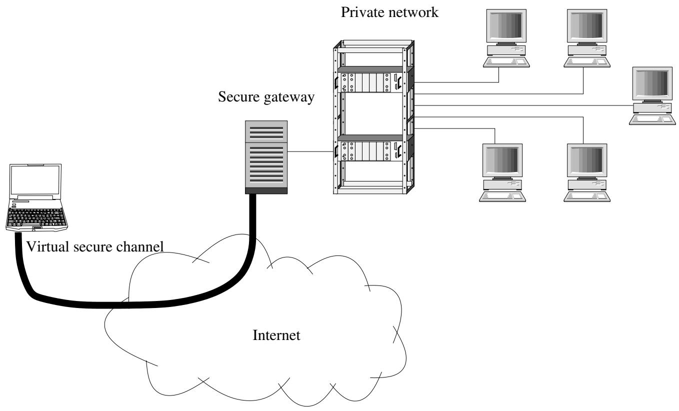

Fig. 1. Security Gateway example: remote user securely connected to a private LAN.

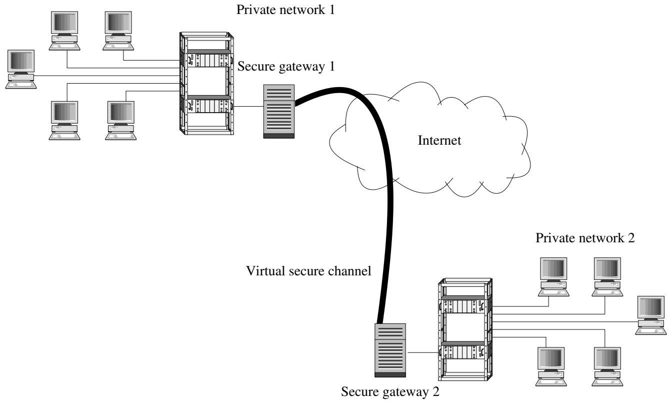

Fig. 2. Security Gateway example: two private LANs connected together through a secure channel.

large data transmissions and for a multimedia application. [9] reports some performance considerations on the FreeS/WAN[10] IPSec implementation. A methodology for estimating the CPU overhead obtained with different IPSec configurations is also reported on this website. [11] reports some results obtained on a 100Mbit/s network by considering the IPSec linux implementation included in 2.6 kernel series[12], [13]. This website only provides results for ESP in transport mode; performance results were obtained by considering the outdated triple DES with a 192-bit key as encryption algorithm and HMAC-SHA-1 as authentication algorithm. In our paper we still provide an evaluation of network performance, but we also consider the CPU usage and effort spent as main parameters. A particular focus is put on the new cryptographic algorithms, AES and SHA-2. Different IPSec configurations are compared and their needs evaluated. The experiments not only take into account the security part of IPSec, but also the IPComp protocol. This protocol is included in IPSec and allows compressing the IP payloads by the means of a compression algorithm [14]. This gives in many cases the possibility (depending on the type of data) to reduce the number of bytes to be sent on the network virtually widening its banwidth.

The target of this study is a secure gateway for low-end market. These machines are gaining importance as the number of network-enabled home devices increases.

In this paper we show the results of the tests we have performed. In Section 2 we provide a description of how these tests were conducted and of the obtained results.

II. IPSEC PERFORMANCE MEASUREMENT

In this section we describe the tests we conducted on an IPSec-based network and we discuss the related results. The aim of these tests was to provide a base from which to evaluate the requirement of IPSec for supporting 100Mbit/s and 10Mbit/s traffic.

As explained in the previous section, our goal was to understand the CPU requirements of different configurations of the IPSec suite of protocols. Comparing different IPSec configurations was not our main goal, but it was done to understand the influence of the different parts (encryption, authentication, compression, . . .) of the protocols. The test we designed is based on sending a long piece of data (1Gbyte) between two PCs.

Here follow a description of the hardware, the software, and the IPSec configurations we used for the tests. The results we obtained by considering a 100Mbit/s and a 10Mbit/s network are then presented.

A. Hardware and software configuration of the test network

The test were done using two PCs running Linux RedHat 7.3 (kernel 2.4.18) patched with the FreeS/WAN 1.99 [10] implementation of IPSec. FreeS/WAN was also patched with the J. Ciarlante's modular algorithm patches (adding support for AES, NULL, and SHA-2 algorithms) [15]. While new Linux kernel releases (2.6) provide a new IPSec implementation, FreeS/WAN was chosen for these tests as it is well known and quite highly optimized for performance. Tests with the new implementation should anyway give results that are not to different from the ones here provided. Linux was chosen both because it is easy to modify (this will be very useful for our future works) and because it provides easy ways of applying measures. The network environment was based on the IPv4 protocol, but further tests will be conducted with IPv6 in the future.

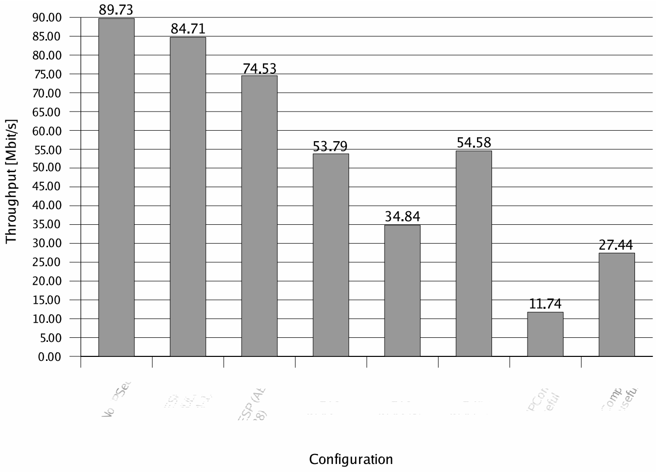

Fig. 3. Network throughput for a 100Mbit/s network.

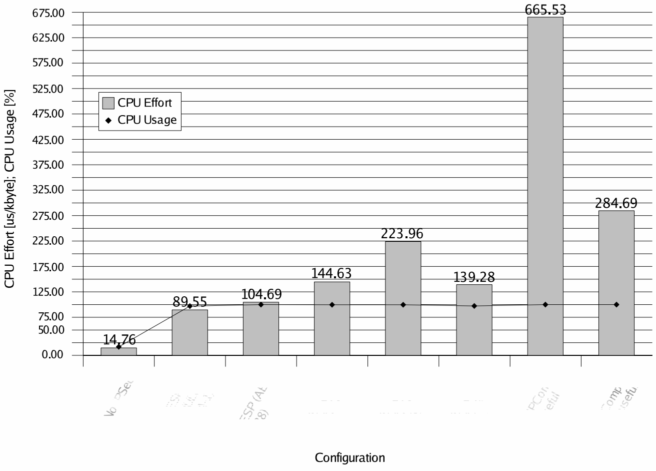

Fig. 4. CPU effort comparison for a 100Mbit/s network. Numbers on the top of the bars represent the CPU effort.

The two PCs we used are 500MHz Intel Pentium III based and were connected to a 100Mbit/s network. While these PCs are to be considered obsolete machines, the results we show here are useful as we are considering small home-gateways as well as embedded systems. The results we obtained are also to be considered partially scalable to larger and more powerful systems.

The tests were partly conducted by the means of the *Netperf* tool [16], a tool for network performance evaluation. A set of Bash [17] scripts [18] were used to take track of instantaneous processor usage and network traffic. The scripts use the information available in the Linux *proc* interface [19]. This interface provides direct access to kernel settings and information.

B. Description of the tests

We chose to use different IPSec configurations in order to be able to represent different usage scenarios and to understand which is the influence of each main part of IPSec on performance. Some of the configurations we chose can be used in real systems, while others are for test only.

For ESP encryption the AES symmetric crypto-algorithm has been selected. So far the Triple-DES algorithm has been the default algorithm used in FreeS/WAN (single DES is the algorithm required for IPSec RFCs conformance). This algorithm is much slower than AES in software (up to 3 times, depending on the implementations) and is an outdated NIST standard for symmetric key cryptography. For ESP authentication the HMAC-SHA-1 algorithm was mainly used even if some tests were conducted by using HMAC-SHA-2 with a 256-bit signature.

The mode always selected for the tests is the tunnel mode, this is because the tests are dedicated to secure gateway machines. In FreeS/WAN it is possible to use the tunnel mode even on host-to-host connections by configuring the ends of the tunnel to be the hosts themselves.

Other tests were conducted by using the IPComp protocol (deflate algorithm). The usage of this protocol was associated to ESP with AES 128-bit encryption and HMAC-SHA-1 authentication. We evaluated the effects of IPComp in two different cases. The first one corresponds to sending data on which compression has a good effect (i.e. the result of the compression operation is shorter than the original data); in the second case a piece of data that cannot be further compressed (a bzip2 file in our case) is sent. In the latter case sent datagrams are not compresses (as they cannot be), but a compressibility test needs to be run. The compressibility test consists of running compression on the payloads and comparing their dimensions with the original ones. Compressed payloads are used when they are smaller then the original ones; they are discarded otherwise.

While it is widely known that the dimension of datagrams has a large influence on the performance of the protocols, we decided not to change this parameter during our tests. As explained before, our focus was not to globally evaluate IPSec performance, but to understand its requirements. Therefore we decided to use the datagram size of 1500byte. This is because we decided not to introduce a further parameter in our analysis which does not directly influence relative results. Furthermore,

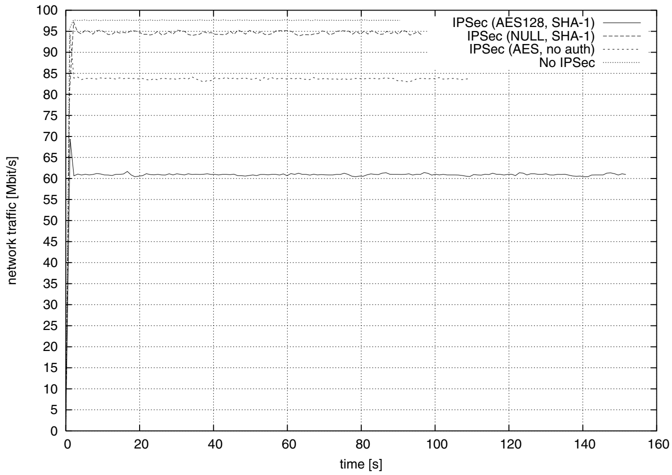

Fig. 5. Network output traffic on the sender PC for a 100Mbit/s network.

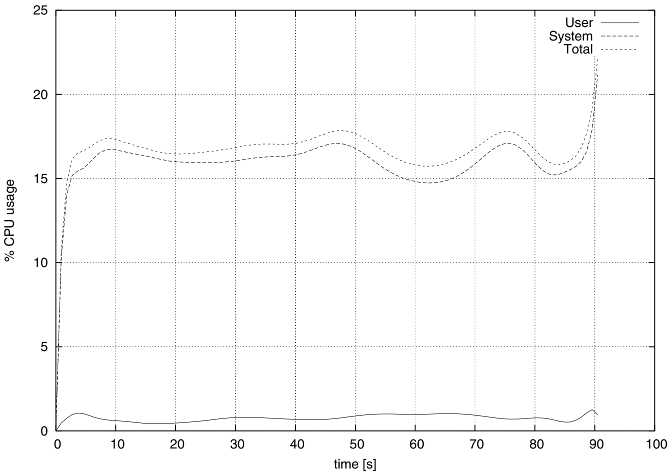

Fig. 6. Instantaneous CPU load for a 100Mbit/s network when IPSec is not enabled.

while the 1500byte packet size represents only the 10% of the packets sizes used during internet communications (the most part of the datagrams is 40byte length only), datagrams of this size carry around the 50% of the whole traffic [20], [21]. Some considerations about packet length and how this parameter influence the usage of crypto-accelerators can be found in [7].

Since a long data set was chosen (1Gbyte) the tests were run only once each. Using a long data set instead of a short one, does not influence performance. In fact Netstat sends an user-defined amount of data by re-sending many times the same packet which is small enough to be held into the main memory. This is done in order not to have the measured performance influenced by the ones of the mass storage devices.

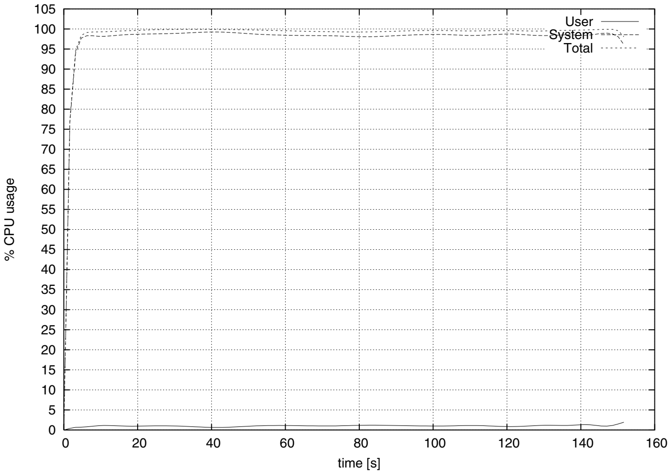

Fig. 7. Instantaneous CPU load for a 100Mbit/s network when IPSec (ESP - AES 128bit - HMAC SHA-1) is enabled.

C. Results

1) 100Mbit/s network: The throughput obtained on a 100Mbit/s network using different IPSec configurations is reported in Figure 3. While in Figure 4 the CPU effort expressed as CPU load in percentage and in time needed to process each sent kilobyte is shown. In all the figures *HMAC-SHA-1* and *HMAC-SHA-2* are respectively shortened to *SHA-1* and *SHA-2*.

Analyzing the reported results, some considerations can be pointed out. While the network capacity is the limiting factor for the network throughput when IPSec is not used, the CPU becomes the limiting factor as soon as IPSec with some form of encryption/authentication is used. Enabling the IPSec ESP protocol in tunnel mode and using authentication-only allows to sustain a data throughput close to the one obtained without enabling IPSec. Unfortunately in this case the CPU usage rises to 97%, more than 5 times the one obtained for the no-IPSec configuration. The authentication-only configuration is rarely used in practice, since it provides no data confidentiality. A configuration that may be usable in real systems – even if is usually believed to be not very safe – is that in which IPSec is used within the ESP protocol in tunnel mode with (AES) encryption enabled and without authentication. In this case the network throughput is decreased by the 17% with respect to the no-IPSec configuration; the CPU usage with this configuration is about 100%. This performance decrease is unacceptable in many cases. Enabling the HMAC-SHA-1 authentication, network throughput dramatically decreases from around 74Mbit/s to 53.8Mbit/s. Changing the authentication algorithm with the new HMAC-

SHA-2, further decreases the network throughput to 34.8Mbit/s, less than half of the available bandwidth. Using the AH authentication associated with the ESP encryption, produces similar results to the ESP encryption plus authentication case. A representation of the output traffic measured on the sender PC for some of the main adopted configurations is shown in Figure 5. The average network traffic is slightly different from the network throughput since it also includes the protocols' headers. It is interesting to analyze the data relative to the CPU time used for processing every sent kilobyte. This gives an idea on the effort needed to process data for each configuration. Encryption only requires an effort that is 17% higher than in the authentication-only case. Encryption and authentication requires an effort that is 66% higher than authentication-only and 38% higher than encryption only.

Introducing IPComp further lowers the network throughput in every case, since, as explained before, at least the compressibility test needs to be executed. When IPComp is useful the throughput lowers to 11.74Mbit/s, 7.6 times slower than in the no-IPSec case and 4.5 times slower than in the IPSec "encryption+authentication(HMAC-SHA-1)" case. This happens even if the total network traffic (composed of data plus protocols' headers) is reduced by 46%. Even in the "not useful compression" case, the network throughput lowers considerably (27.44Mbit/s). This is due to the computational load introduced by the compressability test. The CPU effort is in both cases 665.53μs/kbyte and 284.68μs/kbyte respectively, much higher than all the other previously considered cases.

It is also possible to examine the CPU user and system load distribution for the cases presented above. The CPU load obtained for the "no IPSec" and for the "IPSec - AES - HMAC SHA-1" case are shown in Figure 6 and in Figure 7. From these figures it is evident that only the system CPU load is increasing when IPSec is enabled. This is easy explainable since all the IPSec-related processing is performed in kernel mode.

2) 10Mbit/s network: Some tests were also conducted on a 10Mbit/s network. The network configuration utilized is the same as before, but the sender's bandwidth was limited through the kernel's device queue manager [22]. This is not a very precise method (in fact some little bursts at higher bandwidth are allowed), but we consider it to be precise enough for our evaluations.

In this case the CPU is no longer the limiting factor for the network throughput except in the case when IPComp is enabled and the data to be sent can be compressed.

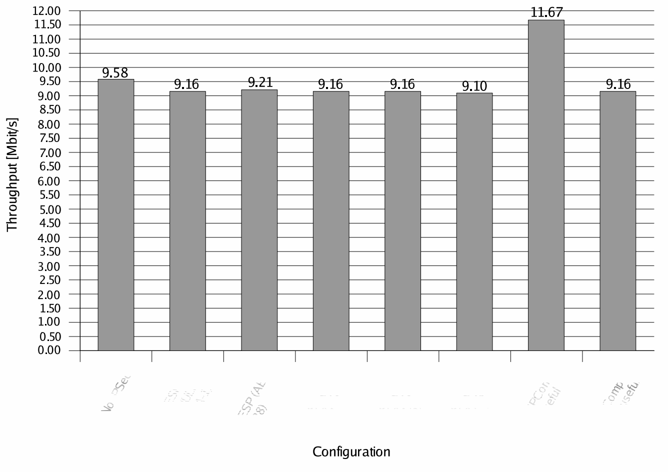

Fig. 8. Network throughput for a 10Mbit/s network.

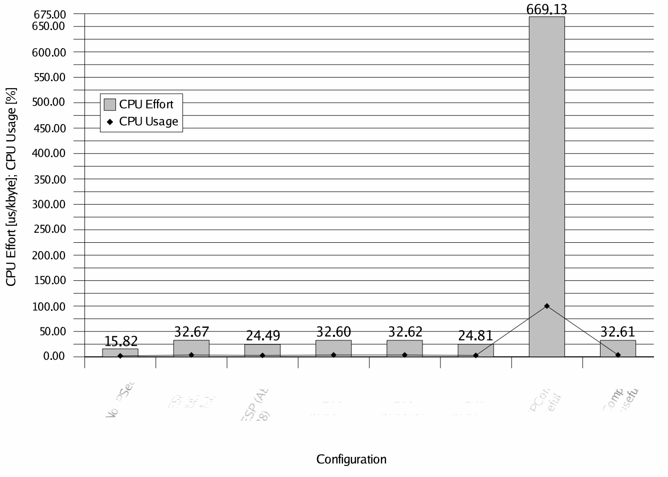

Fig. 9. CPU effort comparison for a 10Mbit/s network. Numbers on the top of the bars represent the CPU effort.

The CPU usage doubles when ESP encryption and authentication is enabled (the CPU utilization is around 2% when IPSec is not used and around 4% when it is used). By enabling IPComp and performing the tests with a file that can be compressed, an interesting result is obtained: the CPU usage goes to 100% so that the CPU become the limiting factor for the network throughput, but the throughput itself rises to 11.67Mbit/s (for a normal 10Mbit/s network without using IPSec we can obtain a maximum throughput of 9.58Mbit/s). This is due to compression that allows for the reduction of the total network traffic by 43% (from 1,070Mbyte to 603Mbyte). If more CPU power were available, the data transfer would be even faster. As a matter of fact 603Mbyte can be transferred in around 481s giving a network throughput of 17.44Mbit/s, around 1.8 times faster than the "no IPSec" case. The network throughput results are shown in Figure 8.

The CPU effort obtained in this case is shown in Figure 9; the effort sustained by the CPU for running IPSec on a slow network is quite low, excluding the case of using IPComp on compressible data. In that case the CPU effort is 42 times the no-IPSec case. While in the 100Mbit/s network case the compressibility test introduced a further slowdown on the network throughput and the CPU effort to rise, in the 10Mbit/s network, both parameters do not change in a noticeable way using this configuration.

III. Conclusions and Future Work

Some evaluations can be done on the results shown above. We need to take into account that, considering the case of an IPSec-based secure gateway, we would also have the computational load due to managing large security association and security policy databases, in addiction to managing the connections and running the VPN server. The same machine will then possibly need to manage firewall rules (if also used as firewall) and routing tables (if also used as a router).

Supposing, as normally done, that Gilder's Law and Moore's Law forecasts are right, the available network bandwidth is growing faster than CPU's computational capacity. Therefore, new strategies for supporting IPSec - and, more in general, secure protocols - need to be studied. When very high network bandwidths are considered, many effects have to be taken into account even if hardware accelerators are used. Often just adding an hardware accelerator is not enough and some hardware/software optimizations have to be put in place to obtain reasonable performance [23].

Even for slower networks some form of hardware acceleration can be a desirable option. As we have seen, using IPComp could allow us to reach considerably higher network throughput (and possibly lower power consumption due to network interface) and this could be really important in limited bandwidth conditions (for example DSL). At the same time, using IPComp can be resource consuming for devices on slow networks (e.g. small embedded systems). If a small IPSec co-processor (including IPComp acceleration) could be added to these devices, their network performance, efficiency, and security could be considerably improved.

While a hardware crypto-accelerator is really key in reaching high performance on big systems, it may also be useful in lowering the CPU usage on small, slow systems.

Some guidelines can be also derived from the results we have obtained. A first consideration should be done on the often used "encrypt everything" policy. As a matter of fact, using encryption when it is not really necessary it is just a resource wasting. In many cases people sending information over the Internet are not really concerned of their privacy, they are just concerned of their authenticity (i.e., being able to verify that data have not being changed during their transmission). Let us think, for example, about a network for collecting air pollution information. This network can be formed by many local measurement equipments sending data to a central database server though the Internet. In this case there is usually no interest in hiding information even though it is important to be be able to verify that the data have not being modified during their transmission. In this case, as in many others, a pure authentication-only policy (implemented, for example, through the ESP protocol with the HMAC-SHA1 algorithm) is enough to provide the necessary protection to data.

When also encryption is required, the correct algorithm need to be chosen to obtain a good level of performance and security. As a matter of fact triple-DES should not be used anymore, both because it has been declared obsolete by NIST and because it is slower than AES. Also the symmetric algorithm key-length selection plays a fundamental role in determining the performance that can be obtained. Using longer key sizes guarantees an higher level of security, but it requires more computational resources. As a matter of fact, 128-bit keys for AES are more than enough to protect most of the present normal communications. For example, the National Security Agency (NSA) has adopted AES for its classified documents: 196 and 256-bit keys are required by them for top-secret level information only [24].

Summarizing, the security-performance trade-off should be carefully evaluated before deploying an IPSec-based system not to waste too much resources without obtaining real benefits from the security stand point.

The IPComp protocol deserves some additional considerations: as shown before, it is very useful in some cases, but also very performance killing in some others. Evaluations on the usefulness of IPComp can be done a priori, by studying the traffic that will be sent over the considered channel. If average packet size is, for example, very small, IPComp should not be used. As a matter of fact, in this case most of the packets will not be compressed and a lot resources will be spent in non-useful compressibility tests. IPComp should also not be used in all the cases in which it is known that higher-level protocols already apply compression on data. As a matter of fact, also in this case a lot of resources can be wasted in non-useful compressibility tests.

A performance measurement in a Mobile IPv6 environment is ongoing. This also includes performance measurement of embedded systems.

References

[1] S. Kent and R. Atkinson, "Security Architecture For the Internet Protocol – RFC2401," IETF RFC, 1998. [Online]. Available: http://www.ietf.org/rfc.html

[2] ——, "IP Authentication Header – RFC2402," IETF RFC, 1998. [Online]. Available: http://www.ietf.org/rfc.html

[3] ——, "IP Encapsulating Security Payload (ESP) – RFC2406," IETF RFC, 1998. [Online]. Available: http://www.ietf.org/rfc.html

[4] D. Harkins and D. Carrell, "The Internet Key Exchange (IKE) – RFC2409," IETF RFC, 1998. [Online]. Available: http://www.ietf.org/rfc.html

[5] J. Feghhi and J. Feghhi, *Secure Networking with Windows 2000 and Trust Services*. Addison Wesley, 2001.

[6] R. Yuan and W. T. Strayer, *Virtual Private Networks*. Addison Wesley, 2001.

[7] S. Miltchev, S. Ioannidis, and A. D. Keromytis, "A Study Of the Relative Costs of Network Security Protocols." Monterey, CA: USENIX Annual Technical Program, June 2002.

[8] S. Ariga, K. Nagahashi, M. Minami, H. Esaki, and J. Murai, "Performance Evaluation Of Data Transmission Using IPSec Over IPv6 Networks," in *INET*, Yokohama, Japan, July 2000.

[9] Performance of FreeS/WAN. FreeS/WAN. [Online]. Available: http://www.freeswan.org/freeswan_trees/freeswan-2.05/doc/performance.ht%ml

[10] FreeS/WAN project. [Online]. Available: http://www.freeswan.org

[11] Vincent Roy. (2004, 11 Oct.) Benchmarks for Native IPsec in the 2.6 Kernel. [Online]. Available: http://www.linuxjournal.com/node/7840

[12] The Linux Kernel Archive. Kernel.Org Organization, Inc. [Online]. Available: http://www.kernel.org

[13] Brian Buesker, Kimmo Koivisto, Bill Nottingham, Christophe Saout, and Ralf Spenneberg. IPSec-Tools. [Online]. Available: http://ipsec-tools.sourceforge.net

[14] A. Shacham, R. Monsour, R. Pereira, and M. Thomas, "IP Payload Compression Protocol (IPComp) – RFC2393," IETF RFC, 1998. [Online]. Available: http://www.ietf.org/rfc.html

[15] J. Ciarlante. Modular algo patches. [Online]. Available: http://ipsecaes.fase.com.br

[16] R. Jones. The Netperf tool. [Online]. Available: http://www.netperf.org

[17] Bash. Free Software Foundation. [Online]. Available: http://www.gnu.org/software/bash/bash.html

[18] Mendel Cooper. (2004, 3 Oct.) Advanced Bash-Scripting Guide. [Online]. Available: http://www.tldp.org/LDP/abs/html/

[19] J. Fink. An overview of the proc filesystem. [Online]. Available: http://www.linuxgazette.com/issue46/fink.html

[20] (1997) WAN packet size distribution. [Online]. Available: http://www.nlanr.net/NA/Learn/packetsizes.html

[21] S. McCreary. Packet length distribution. [Online]. Available: http://www.caida.org/analysis/AIX/plen_hist/

[22] M. A. Brown. (2003, September) Traffic control howto. [Online]. Available: http://linux-ip.net/articles/Traffic-Control-HOWTO/

[23] E. P. Markatos, "Speeding up TCP/IP: Faster processors are not enough," Foundation for Research & Technology - Hellas (FORTH), Tech. Rep., 2002.

[24] (2003, June) CNSS Policy No. 15, Fact Sheet No. 1 National Policy on the Use of the Advanced Encryption Standard (AES) to Protect National Security Systems and National Security Information. CNSS. [Online]. Available: http://www.nstissc.gov/Assets/pdf/fact%20sheet.pdf