

The Performance Evaluation of a Dynamic Configuration Method over IPSEC

[†]Shou-Heng Liu, [‡]Yung-Feng Lu, Chin-Fu Kuo, Ai-Chun Pang, Tei-Wei Kuo

Department of Computer Science and Information Engineering

[†]Department of Electrical Engineering

National Taiwan University, Taipei, Taiwan

[‡]Institute for Information Industry, Taiwan

IPSEC provides the network layer additional security functionality to transform multiple segments of a private network into one. However, the setup of IPSEC-based secured tunnels tends to be static in the configuration and restricted in a machine-to-machine fashion. This paper is motivated by the needs of users in moving among working environments while maintaining a secured connection to its primary working environment. Instead of adopting a full-scaled PKI, we propose to extend IPSEC with a more light-weighted and flexible support on user mobility. The system overheads and performance of IPSEC were explored based on the Common Criteria (CC), an international standard in the definitions of the information technology security requirements [7].

Keywords: IPSEC, Security, Common Criteria, Performance

1. Introduction

The Virtual Private Network (VPN) is an extension of a private network that encompasses network links across public networks (e.g., Internet). Within a VPN, authorized users are allowed to access various network-related services and resources. However, security-enhanced services usually come at the price of additional overheads and might reduce the system performance. While excellent work is done for real-time communication, e.g., [1, 11], little work is done in exploring real-time secured communication. How to balance the system security level and the resulted system performance relies on the understanding of security technology and the targeting threats. The purpose of this paper is motivated by the investigation of a popular VPN protocol and its impacts on system resource usages.

The Point-to-Point Tunneling Protocol (PPTP), the Layer 2 Tunneling Protocol (L2TP), and the IP Security Protocol (IPSEC) are the main tunneling technologies adopted for VPN. PPTP and L2TP operate on the data-link layer and are based on the Point-to-Point Protocol (PPP). VPN gateways with PPTP or L2TP have to possess the PPP capability and usually come with more overheads. On the other hand, IPSEC operates directly on IP packets without an additional layer for data transmissions between gateways. Prior to the proposing of IPSEC, network operators were forced to deploy solutions such as those at the application level, e.g., Secure Sockets Layer

(SSL). Compared with IPSEC, SSL provides less flexibility and is less efficient. IPSEC was proposed by the Internet Engineering Task Force (IETF) to provide the network layer additional security functionality. A certain degree of communication security against eavesdropping, repudiation, and spoofing is provided. Although IPSEC introduces a good flexibility in the security enhancement of higher-layer protocols, security-enhanced services comes at the price of additional overheads. The design of IPSEC also makes itself more suitable to the establishment of machine-to-machine secured tunnels.

The Common Criteria (CC), that is an international standard in the definitions of the information technology security requirements [7], serves as a good roadmap in exploring tradeoffs between system performance and security. The purpose of this paper is to first exploit the relationship between security objectives of CC and IPSEC such that threats against IPSEC could be better understood. We then extend IPSEC to provide a lower-cost and flexible way in joining a virtual private network, compared to a full-scaled Public Key Infrastructure. This research is motivated by the needs of users in moving among working environments while maintaining a secured connection to its primary working environment. We propose a key-based algorithm to extend IPSEC for a more dynamic way in setting up secured tunnels. The implementation of the proposed algorithm is simple and highly portable. The tradeoffs between performance/overheads and security objectives are explored based on performance measurements in [8, 9, 10]¹ and experiments over IPSEC with different selected options and encryption/decryption algorithms.

The rest of this paper is organized as follows: Section 2 explores the relationship between IPSEC and CC. We then propose an IPSEC-based algorithm to dynamically set up secured tunnels with different security considerations. Section 3 provides discussions on the implementation issues for the proposed algorithm and performance measurements with respective to different security objectives. Section 4 is the conclusion.

2. IPSEC with Dynamic Configuration Supports

2.1 IPSEC and Threats

IPSEC encrypts higher-layer protocols by having an authentication header (AH)

¹ Lin, et al [3,4] shows that the throughput of downloading files for authentication header (AH) is very close to that for encapsulating security payload (ESP) (with null encryption) with the same authentication algorithm.

and an encapsulated security payload (ESP). With an AH, the strong integrity of an IP datagram and its authentication and non-repudiation could be provided. The ESP of an IPSEC packet provides confidentiality, data origin authentication, connectionless integrity (excluding the IP header fields), an anti-replay service (a form of partial sequence integrity), and limited traffic flow confidentiality [5]. The primary difference between authentications provided by ESP and AH is the extent of the coverage. Furthermore, IPSEC uses public and private keys to encrypt and decrypt transferred data. Before any tunnel establishment, two devices for a tunnel need to share the keys. The Internet Key Exchange (IKE) [6] negotiates connection parameters, including keys, for AH's and ESP's.

Although IPSEC introduces a good flexibility in the security enhancement of higher-layer protocols, security-enhanced services comes at the price of additional overheads. Before the exploring of the extra overheads issues due to IPSEC, we shall first identify the potential threats and the corresponding security objectives targeted by IPSEC. The Common Criteria (CC), that is an international standard in the definitions of the information technology security requirements [7], serves as a good vehicle in exploring the tradeoff between performance/overheads and security.

Security Objective	Threat
O.Crypto_Comm_Channel	T.Spoofing
O.Data_Exchange_Conf	T.Hack_Comm_Eavesdrop
O.Comm_Trusted_Channel	T.Spoofing
O.Integrity_Attr_Exch	T.Spoofing
O.Non-repudiation	T.Spoofing

Table 2-1: The mapping of security objectives and threats for IPSEC

Two threats in CC match the security targets of IPSEC: *T.Hack_Comm_Eavesdrop* and *T.Spoofing*. *T.Hack_Comm_Eavesdrop* concerns situations in which hackers obtain user data by eavesdropping on communications lines. *T.Spoofing* considers situations in which attackers trick users into interacting with spurious system services. To be more specific, security objects corresponding to the above two threats are listed in Table 2-1 [7]. Under *O.Crypto_Comm_Channel*, a secure session establishment should be provided between the system and remote systems using encryption functions. With *O.Data_Exchange_Conf*, user data confidentiality should be protected when data are exchanged between systems. *O.Comm_Trusted_Channel* concerns the provision of a communications channel

between the system and a remote trusted system for security-critical operations. *O.Integrity_Attr_Exch* is to ensure correct exchanging of attributes with another trusted product. It ensures that the system correctly exchanges security-attribute information with another trusted IT product. *O.Non-repudiation* is to prevent users from avoiding accountability. With security objectives, we could better point out the security enhancement features that IPSEC provides and explore more complex and advanced services over IPSEC. Tradeoffs between performance/overheads and security objectives could be better understood.

IPSEC Option	O.Comm_Trusted_Channel	O.Data_Exchange_Conf	O.Crypto_Comm_Channel	O.Integrity_Attr_Exch	O.Non-repudiation
Native	-	-	-	-	-
AH	-	-	-	●	●
ESP	●	●	●	-	-
AH + ESP	●	●	●	●	●

Native: no security service at all, AH: packets with the AH header, ESP: packets with the ESP format

Table 2-2: Mapping of security objectives and IPSEC options

IPSEC can be configured into three operating modes with different security functionalities. The mapping of security objectives and IPSEC options is summarized in Table 2-2. The AH feature satisfies *O.Integrity_Attr_Exch* and *O.Non-repudiation*. The ESP feature serves the requirements of *O.Comm_Trusted_Channel*, *O.Data_Exchange_Conf*, and *O.Crypto_Comm_Channel*. In the next section, we shall propose a user mobility service over IPSEC and then present the tradeoffs between performance/overheads and security objectives in later sections.

2.2 A Dynamic Extension of IPSEC

The purpose of this section is to further extend IPSEC to provide a lower-cost and flexible way in joining a virtual private network (VPN). It is motivated by the needs of users in moving among working environments while maintaining its connection to its primary secured working environment. While IPSEC provides a good way in merging segments of VPN's into one, the setup itself tends to be more static in the configuration or restricted in a machine-to-machine fashion.

Consider a case in which Alice must go to a client's office and discuss with her

client regarding various classified information. Let the information stay somewhere at her primary working environment, e.g., her company's archive department. Suppose that her client's office and her primary working environment are not in the same VPN. Since the discussion is only a single event, and it is only for Alice's one-time visiting, it is not realistic to have permanent secured networking between some machine of Alice's client and her primary working environment. Some tunnel should be set up dynamically only for Alice and merely fits her needs at this moment. Instead of adopting a full-scaled Public Key Infrastructure, the purpose of this section is to propose an IPSEC-based algorithm for better support on user mobility. We shall then further exploit the performance issues for the security objectives of IPSEC in a later section.

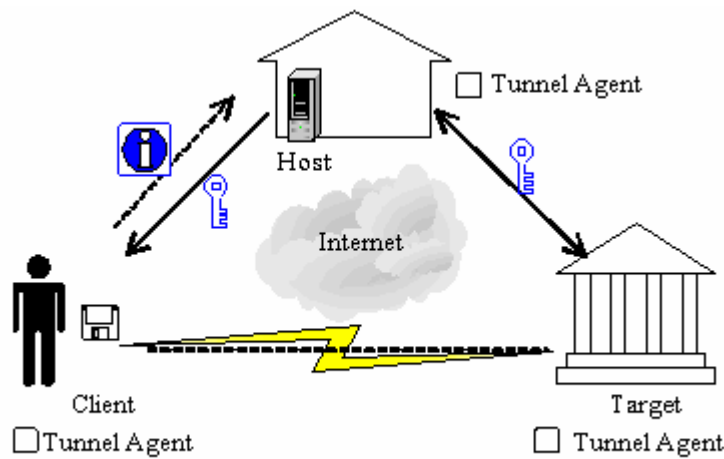


Figure 2-1: System Architecture

As shown in Figure 2-1, *Client*, *Host*, and *Target* denote the moving subject (i.e., Alice in the previous example), a trusted machine of the subject (e.g., a machine on the desk of Alice), and a trusted machine of the primary working environment of the subject (e.g., a machine in the company's archive department), respectively. The objective is to provide a secured IPSEC-based tunnel between Client and Target dynamically. We assume that Host and Target could communicate in a trusted way (Note that when it is a concern, IPSEC could be deployed to resolve the problem). In the algorithm, we use a key-based mechanism to achieve the objective. Let Client and Host both have the same set of private and public keys. When Client requests a secured tunnel between herself and Target, Client encrypts the following request with the public key of Client and submits it to Host:

Request ID || User ID || Objective-ID || Request Info || Authentication Tag

Request ID, User ID, and Request Info are a uniquely identifiable ID for the request, the ID of Client, and other necessary information for the request, respectively. Objective-ID denotes the security objectives for the tunneling, e.g., those corresponding to the four rows in Table 2-2. Authentication Tag is the request's digital signature, that is the request encrypted with the private key of Client. The establishment of a secured tunnel between Client and Target could be done as follows:

【Step1: Initial Setup Request: Client to Host】

Client uses her Tunnel Agent to submit a request to Host to request for a secured connection.

【Step2: Ticket Validation: Client】

The Tunnel Agent of Host uses Client's private key to decrypt the request and then uses Client's public key to verify the digital signature of the request.

【Step3: User Authentication at Target: Host to Target】

The Tunnel Agent of Host passes user information to Target. The Tunnel Agent of Target authorizes the user to use services provided by Target.

【Step4: Establishment of Session Key: Target】

The Tunnel Agent of Target establishes a new tunnel and generates a session key for the tunnel. The session key and the tunneling information are passed back to the Tunnel Agent of Host.

【Step5: Delivery of Session Key and Tunneling Information to Client: Target to Host】

The Tunnel Agent of Host encrypts the following response message (including the tunneling information and session key) with the public key of Client and then sends it back to Client:

Request ID || Session Key || Tunneling Info || Authentication Tag

Request ID, Session Key, and Tunneling Info are the request ID, the session key, and tunneling information, respectively. Authentication Tag is the response message's digital signature, that is the request encrypted with the private key of Client.

【Step6: Establishment of a Secured Tunnel: Client and Target】

The Tunnel Agent of Client decrypts the information with the private key of Client and establishes a tunnel with Target.

Note that the implementation of the above key mechanism could be done with some piece of hardware containing the private and public keys.

3. Implementation and Performance Analysis

3.1 Implementation Remarks

The implementation of the IPSEC-based algorithm proposed in Section 2.2 should consider the security objectives for each establishment of a secured tunnel and its impacts on the system performance. Each tunnel establishment might consider a combination of three kinds of options: (security objectives, IPSEC options, encryption/decryption algorithms) The set of selected security objectives denote the security level expected by users, and there is a mapping between security objectives and IPSEC options (such as AH, ESP, AH+ESP), as shown in Table 2.2.

There are different encryption/decryption algorithms for IPSEC implementations, and each of them represents different system overheads and security strength (we shall provide analysis and performance evaluation in Section 3.2). SHA-1 and MD5 are alternatives to provide authentication functions for AH and ESP. DES and 3-DES are options for ESP to encrypt the payloads of IP packets. Different levels of security and performances are provided for different algorithms. Take brute-force attacks on encrypted information as an example. Any attempt in producing two messages with the same digest needs the order of 2^{64} operations for MD5 and 2^{80} operations for SHA-1, respectively. SHA-1 is considerably stronger against brute-force attacks, compared to MD5 [3]. However, SHA-1 (160-bit) needs a larger buffer than MD5 (128-bit) does and has more overheads.

The implementation of the proposed IPSEC-based algorithm is simple and highly portable. Consider the system architecture, as shown in Figure 2-1: The responsibilities of Tunnel Agents of Client/Host/Target are for the encryption and decryption of requests/responses and the construction/destruction of secured tunnels. The construction and destruction of secured tunnels could be done by proper modifications to a configuration file for IPSEC (e.g., ipsec.conf in an IPSEC implementation *FreeS/WAN*). Tunnel Agents could trigger the construction and destruction of tunnels with IPSEC commands in the shell program. As a result, Tunnel Agents could be implemented pretty independently from many IPSEC implementations. That is, Tunnel Agents can be implemented without modifications to

IPSEC implementations, e.g. FreeS/WAN [8].

We must point out that although the proposed IPSEC-based algorithm consists of Client, Host, and Target, they are merely three logical objects. In other words, Host and Target could be located at the same physical computer. The advantage for the three logical objects is on the flexibility for different needs in system designs. The tradeoff is apparently on extra message delivery and coding/decoding efforts.

3.2 Performance Analysis Evaluation

The purpose of this section is to explore the system overheads of IPSEC-based mechanisms and the usage of system resources, such as CPU. We first present a performance summary report derived from results in the literature [8, 9, 10]. We then conclude this section by a performance evaluation of IPSEC with different selected options and encryption/decryption algorithms.

Although the proposed IPSEC-based algorithm consists of tunnel setups, key exchanges, and transmissions of data packets, the overheads of the proposed algorithm is dominated by the encryption/decryption and authentication operations for data packets, provided that a significant amount of information flows between Client and Target. In this section, we focus our performance evaluation on IPSEC with different selected options and encryption/decryption algorithms, where different selections of IPSEC options reflected different needs of security objectives (Please see Table 2-2).

	Algorithms	Overheads (Unit: Cycles/Byte)
Authentication	MD5	5
	SHA-1	13
	RIPEMD-160	16
Payload Encryption	DES	43
	3-DES	116
	AES	58

Table 3-1: Overheads for different encryption/decryption algorithms

With the popularity of IPSEC, there have been many performance evaluations on IPSEC in the past years. Instead of redoing all experiments, we first derived a performance summary report for IPSEC implementations in the past work. Note that

many performance evaluations were done over different platforms, and a significant efforts was needed in merging results. Table 3-1 shows the overheads, due to header authentication and payload encryption, where the overheads were measured in terms of CPU cycles needed for each byte transmission. Note that packet transmissions with IPSEC installed but without any selected encryption/decryption services would consume CPU cycles already. For example, the NAI Labs reported that 6 cycles/byte would be the existing overheads for the case (excluding the network overheads) [9]. In other words, with SHA-1 selected, the total overheads would be 19 cycles/byte. The overheads for authentication in Table 3-1 were mainly based on the excellent measurement work done at the NAI Labs for FreeS/WAN [9], where Pentium II was the major experimental platform. Other measurement work was done by Bosselaers [10] for SHA-1, MD5, and RIPEMD-160, and similar results were reported. The overheads for payload encryption in Table 3-1 were mainly based on results in [8, 9]. It was shown that FreeS/WAN with AES selected roughly doubled the IPSEC throughput, compared to that of FreeS/WAN with 3-DES [9].

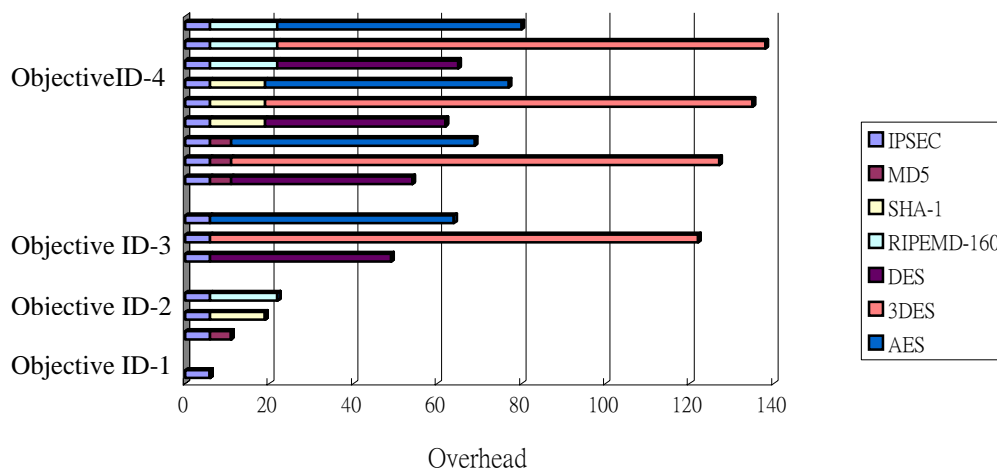


Figure 3-1: Security objectives versus IPSEC overheads

Figure 3-1 shows the system overheads with respect to different security objectives, where each ID-*x* was Objective-ID for the tunneling. That is, ID-*x*'s corresponded to the four rows in Table 2-2. As shown in Figure 3-1, each Objective-ID had the basic overheads, which were for the overheads of packet transmissions with IPSEC installed but without any selected encryption/decryption services. For each Objective-ID, different system overheads were reported for different encryption algorithms. The results were derived from the excellent measurement work in [8, 9, 10].

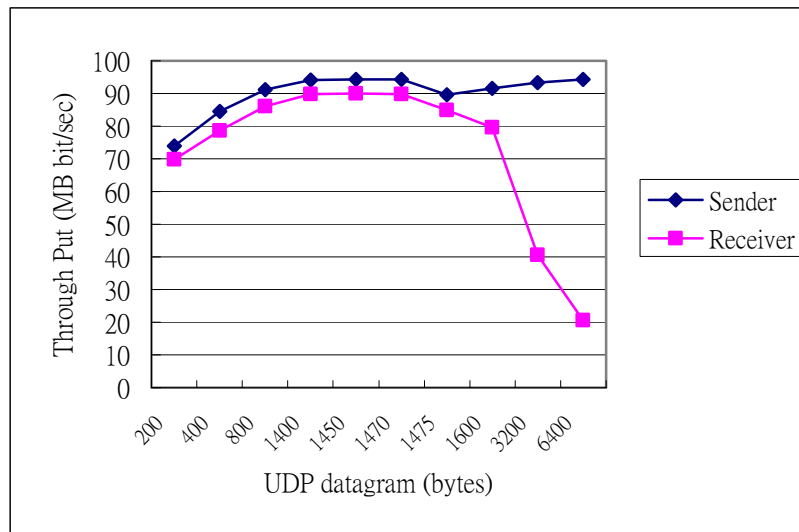


Figure 3-2: Network throughput under different UDP datagram sizes without IPSEC

The second part of the performance evaluation was done by measuring the system performance and system resource usages for data transmissions between two Pentium-IV 1.7GHZ platforms with Linux 2.4.18-14 and FreeS/WAN 2.01. We adopted 100Mbps Ethernet LAN for performance evaluations. A utility iperf was used to do performance evaluations.

Figure3-2 shows the network throughput under different UDP datagram sizes without IPSEC. It was shown that the throughput for the receiver dropped rapidly when the datagram size was approximately over 1470B. The phenomenon was due to the maximum IP packet size (1518B) [12] and the efforts of the receiver in merging IP packets (because of the splitting of over-sized packets). The results served as a base to observe the behavior of IPSEC.

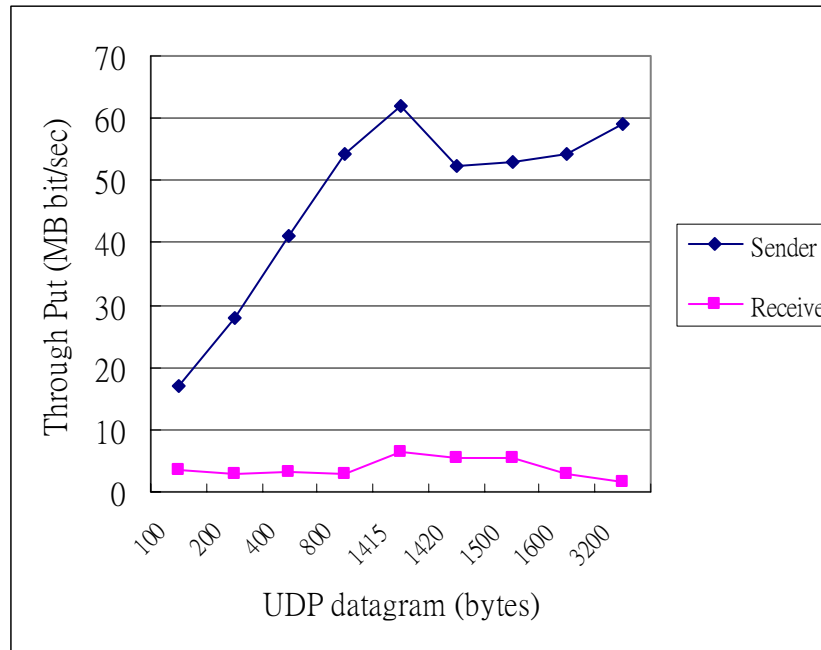


Figure 3-3: Network throughput under different UDP datagram sizes with IPSEC (ESP: MD5+3DES)

Figure 3-3 shows the network throughput under different UDP datagram sizes with IPSEC (ESP: MD5+3DES). We took IPSEC (ESP: MD5+3DES) for evaluations because it was one of the most popular settings for IPSEC. Since the network throughput reaches 20Mbps, the CPU utilization approached 100%. When the datagram size increased, the throughput kept increasing until the peak (1415B) because datagram sizes before the peak could well fit in the ESP of an IP packet. The dropping of the network throughput after the peak was because of the splitting of over-sized IP packets. The restoring of the network throughput after the peak had the same reason why the network throughput increased before the peak. The throughput for the receiver was very low (only few megabits per second) probably because of the limitation of buffer sizes and the resulted delaying on decryption work.

Model	Throughput (MB bits/s)	System Time Usage (%)	Overheads (Cycles/Byte)
Native	92.6	9.9	14.5453
AH-MD5-96	91.0	36.9	55.0922
AH-SHA1-96	91.2	60.4	90.0094
ESP-MD5-3DES	53.9	92.0	231.9884

Table 3-2: System overheads for the peak throughput over IPSEC (TCP) for different encryption algorithms.

We evaluated the system overheads for IPSEC with TCP and UDP. Because of

similar results, we only summarize results for IPSEC with TCP in Table 3-2. The maximum throughputs for different encryption functions were presented. We must point out that the results in Table 3-1 only cover the encryption cost, while results in Table 3-2 include encryption cost and data transmissions. Note that the CPU utilization of the entire system was approximately 100%. The system time usages shown in Table 3-2 only reveal the CPU utilization in terms of the kernel.

4. Conclusion

This paper investigates the system overheads/performance of IPSEC implementations and the targeting security objectives. It provides a foundation in the exploring of tradeoffs between the system security level and the resulted system performance. IPSEC is studied based on the targeting threats and security objectives of the Common Criteria (CC), an international standard in the definitions of the information technology security requirements [7]. We extend IPSEC to provide a lower-cost and flexible way in joining a virtual private network, compared to the Public Key Infrastructure. Secured tunnels could be set up dynamically and automatically based on the proposed key-based algorithm with different security objectives. The implementation of the proposed algorithm is simple and highly portable. We explore the tradeoffs between performance/overheads and security objectives based on previous results [8, 9, 10] and realistic system evaluations. It was observed that different security objectives on IPSEC result in very different impacts on system performance.

In the performance evaluation, the CPU usage of experimented implementations was often close to 100% when some encryption/decryption functions were selected for IPSEC, and the network load was more than 20Mbps. For future research, we shall explore the impacts of IPSEC implementations on user-level real-time applications (such as streaming systems) when severe system resource competitions occur. We shall also investigate a lower-cost implementation of IPSEC to support light-weighted embedded systems.

References:

- [1] Byung-Kyu Choi, Dong Xuan, Riccardo Bettati, Wei Zhao, Chengzhi Li, "Utilization-Based Admission Control for Scalable Real-Time Communication", *Real-Time Systems* 24(2): 171-202 (2003)

- [2] Cisco White Paper - IPSEC
http://www.cisco.com/warp/public/cc/so/neso/sqso/eqso/ipsec_wp.htm
- [3] Jin-Cherng Lin, Ching-Tien Chang, Wei-Tao, "Design, Implementation and Performance Evaluation of IP-VPN", *17 th International Conference on Advanced Information Networking and Applications (AINA'03)*, March 27 - 29, 2003.
- [4] G. C. Hadjichristofi, N. J. Davis, and S. F. Midkiff, "IPSec Overhead in Wireline and Wireless Networks for Web and Email Applications," *IEEE International Performance, Computing, and Communications Conference*, Phoenix, AZ, April 9-11, 2003, pp. 543-547.
- [5] RFC 2406 - IP Encapsulating Security Payload (ESP)
- [6] D. Harkins, D. Carrel. "The Internet Key Exchange." Internet Request for Comments RFC 2409, November 1998.
- [7] Common Criteria for Information Security Evaluation, Version 2.1. CCIMB-99-031. August 1999.
- [8] Linux FreeS/WAN, <http://www.freeswan.org/>
- [9] Adaptive Cryptographically Synchronized Authentication (ACSA) Final Report, http://www.networkassociates.com/us/_tier0/nailabs/_media/research_projects/cryptographic/acsa_final_report.pdf
- [10] Antoon Bosselaers, Fast Implementations on the Pentium
<http://www.esat.kuleuven.ac.be/~bosselae/fast.html>
- [11] M. Adamou, S. Khanna, I. Lee, I. Shin and S. Zhou, "Fair Real-time Traffic Scheduling over A Wireless LAN", *Proceedings of the 22nd IEEE Real-Time Systems Symposium*, RTSS 2001.
- [12] RFC 1042 – Standard for the transmission of IP datagrams over IEEE 802 networks.