# 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 106, ROC
[+]Institute for Information Industry, Taipei, Taiwan 106, ROC

**Abstract**

*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. 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) [7].*

**Keywords:** IPSEC, Security, Common Criteria, Performance

## 1. Introduction

Within a Virtual Private Network (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], 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. VPN gateways with PPTP or L2TP have to possess the PPP capability and usually come with more overheads. IPSEC operates directly on IP packets without an additional layer for data transmissions between gateways. Compared with IPSEC, Secure Sockets Layer (SSL) provides less flexibility and is less efficient [2]. 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 come at the price of additional overheads. The design of IPSEC also makes itself more suitable to the establishment of machine-to-machine secured tunnels.

*Supported in part by research grants from the Institute for Information Industry, the National Science Council under Grant NSC92-2213-E-002-901 and NSC-92-2213-E-002-092, and Microsoft.

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 VPN. 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) 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. The Internet Key Exchange (IKE) [6] negotiates connection parameters, including keys, for

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

AH's and ESP's.

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 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 (O1) | T.Spoofing |
| O.Data_Exchange_Conf (O2) | T.Hack_Comm_ Eavesdrop |
| O.Comm_Trusted_Channel (O3) | T.Spoofing |
| O.Integrity_Attr_Exch (O4) | T.Spoofing |
| O.Non-repudiation (O5) | 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. 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.

| IPSEC Option | **O1** | **O2** | **O3** | **O4** | **O5** |
|---|---|---|---|---|---|
| Native | - | - | - | - | - |
| AH | - | - | - | X | X |
| ESP | X | X | X | - | - |
| AH + ESP | X | X | X | X | X |

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

## 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 VPN. 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. 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.

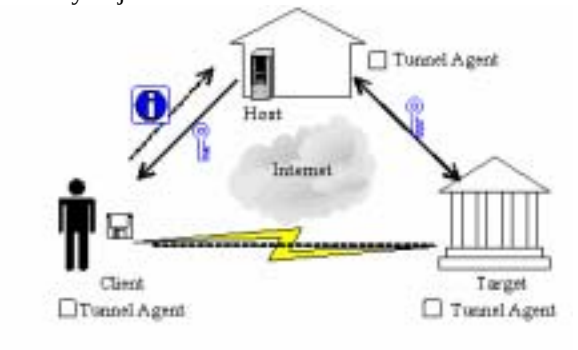


**Figure 2-1**: System architecture

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 [3] 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.

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.

| | **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 [8, 9, 10]

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).

Note that many performance evaluations on IPSEC were done over different platforms, and a significant effort 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. 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]. 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].

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].

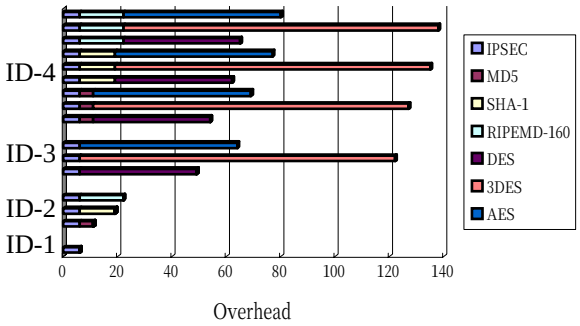


**Figure 3-1**: Security objectives versus IPSEC overheads

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 [4] was used to do performance evaluations.

In the experiments, the network throughput under different UDP datagram sizes without IPSEC was measured. 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) [11] 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.

When the network throughput under different UDP datagram sizes with IPSEC (ESP: MD5+3DES) was considered, 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 of the CC [7]. It provides a foundation in the exploring of tradeoffs between the system security level and the resulted system performance. We extend IPSEC to provide a lower-cost and flexible way in joining a virtual private network, compared to the Public Key Infrastructure. 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.

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", *17th International Conference on Advanced Information Networking and Applications (AINA'03)*, March 27 - 29, 2003.
[4] iperf, http://dast.nlanr.net/Projects/Iperf/
[5] RFC 2406 - IP Encapsulating Security Payload (ESP).
[6] RFC 2409 - The Internet Key Exchange (IKE).
[7] Common Criteria, http://www.commoncriteria.org/
[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] RFC 1042 – Standard for the transmission of IP datagrams over IEEE 802 networks.