Evaluation of IPsec Overhead for VoIP using a Bare PC

N. Kazemi, A. L. Wijesinha, and R. Karne
Department of Computer and Information Sciences
Towson University
Towson, MD 21252
nkazemi@adabtek.com, {awijesinha, rkarne}@towson.edu

***Abstract*— In a bare PC softphone, the VoIP application runs directly on the hardware without any Operating System (OS). Such softphones are useful when security and/or performance concerns outweigh the need for a conventional system. We evaluate the overhead of IPsec for VoIP in a small test LAN using a bare PC softphone The results show that 1) for incoming packets, the difference between the processing time in tunnel and transport modes is about 0.02 ms regardless of voice packet size; 2) the processing overhead increase percentage for incoming packets compared to outgoing packets is largest for ESP without authentication in tunnel mode, and smallest for ESP with either authentication option in transport mode; 3) if UDP processing is eliminated, a throughput increase between 6-14% may be achieved for incoming packets; 4) the throughput increase on a bare PC softphone relative to an OS-based softphone is between 8-15%, and the difference due to varying the voice packet size (except for 40 ms packets) is less than 3%; 5) intrinsic jitter on a bare PC softphone is negligible, and maximum delta (packet inter-arrival time) on a compatible Linux softphone is twice that on a bare PC softphone.**

Keywords-IPsec; VoIP; network security; bare machine computing; bare PC; application object; performance

I. INTRODUCTION

In bare PC computing [1], applications execute directly on the hardware without an Operating System (OS). A portable device, such as, a USB memory stick, contains the complete executable code for one or more applications (no hard disk is used). Bare PC computing is characterized by reduced overhead and improved resistance to attacks targeting vulnerabilities in conventional OSs. A bare PC is also useful for measuring the intrinsic performance of an application or device when the overhead and effects of a conventional OS are eliminated. In the case of a network application, direct execution on the hardware allows novel protocol optimizations such as protocol intertwining to be tested. In particular, since a bare PC has no OS, the performance and overhead of a security component can be precisely measured to provide a bound on what may be achieved with a streamlined or customized conventional OS with minimal functionality.

In this paper, we measure the overhead of IPsec for VoIP and its impact on voice quality using bare PC softphones. We first examine the performance differences due to different cryptographic algorithms, and compare VoIP call quality measurements using bare PC and Linux softphones. We then study two protocol optimizations for bare PC VoIP that eliminate the UDP header and enable direct communication between the UDP and IPsec handlers. The rest of this paper is organized as follows. In Section II, we provide a brief overview of bare PC computing, the design of VoIP with IPsec for a bare PC, and related work. In Section III, we present the results of experiments to measure the overhead of IPsec on the bare PC VoIP application, its impact on call quality, and the performance improvements due to bare PC optimizations. Section IV contains the conclusion.

II. BACKGROUND

A. Bare Machine Computing

In a bare PC, an Application Object (AO) [2] contains code that enables one or more applications to directly execute on the hardware without using an OS. An AO optimizes CPU, memory usage, and task scheduling, and eliminates data copying by directly accessing link layer (e.g., Ethernet) buffers. It also includes streamlined versions of network protocols that are intertwined with the application and facilitate cross-layer communication, and drivers needed by the application to communicate with the audio chip/card or network interface. AOs are very efficient since they are designed with minimal functionality to serve the needs of specific servers or client applications. Details of a hardware API used by bare PC applications are given in [3].

B. IPsec on a bare PC softphone

IPsec [4] is primarily used for securing IP traffic in Virtual Private Networks (VPNs). The Encapsulating Security Payload (ESP) [5] is an IPsec protocol that provides data-confidentiality (encryption) service, and optionally data-origin authentication and data-integrity services. ESP may also be used to protect against packet replay attacks and for traffic flow confidentiality. In transport mode (TR), IPsec protects the IP payload, whereas in tunnel mode (TU), the entire IP datagram is protected. In the latter mode, which is used in VPNs, a security gateway or router implements IPsec at the ends of a tunnel and protects all tunneled traffic. A bare PC softphone with IPsec uses IKEv2 [6] for managing keys and establishing security associations. Since our focus in this paper is on the impact of IPsec on voice quality, we do not study performance of IKEv2 and SIP, which may be used for call setup.

C. Related Work

A study of IPsec in a VPN using videoconferencing traffic [7] reported that delay, jitter, and quality were within acceptable limits for moderate traffic, but not for heavy traffic.

In [8], QoS components were added to an IPsec implementation on Linux systems, and low delay, jitter, and packet loss were achieved in stress tests using artificially generated traffic loads. In an early study [9], packet inter-arrival times (delta) were measured in a test network using IP-sec tunneled voice traffic between two phones, which competed with simulated ordinary traffic. All streams were encrypted using triple-DES, and the results showed that prioritized voice traffic had the most stable inter-arrival times. The difference between previous work and our study is that we measured IPsec overhead for VoIP using a novel bare PC softphone with no OS, and compared values of intrinsic call quality parameters with a modern Linux softphone. The design of a secure bare PC softphone with IPsec in this study extends the design of a bare PC softphone without any security mechanisms described in [10]. It also adapts the design of IPsec for bare PC web and email servers over TCP given in [11] to work with VoIP over RTP/UDP.

III. RESULTS

We conducted several experiments to measure IPsec overhead on a bare PC softphone, and to compare VoIP performance with IPsec using a bare PC and a compatible Linux softphone. The experiments used a dedicated test LAN as described below. The actual IPsec overhead throughput on a bare PC softphone was determined by measuring the individual processing times for incoming and outgoing voice packets.

A. Experimental Setup

We made calls between a pair of bare PC softphones and a pair of Linux softphones, using ESP in transport or tunnel mode for security. Although tunnel mode is typically implemented on VPN gateways, we evaluated tunnel mode as an alternative to transport mode for end-to-end VoIP security. The softphones were directly connected by a 100 Mbps Cisco FastHub 400 series. Each PC running bare PC softphone had a 2.4 MHz CPU and 512 MB RAM, while each Linux PC had 3.2 MHz CPU with 1024 MB RAM and ran Linphone [12] Version 2.1.1 on Fedora 10 Kernel Version 2.6.27.5-117-fc10.i686. We used Wireshark [13] version 1.2.1 on a Windows XP machine with a 2.4 MHz CPU and 1024 MB RAM to capture packets during a call and provide VoIP performance statistics. Bare and Linux PCs included 3COM905CX 10/100 and IntelPRO/100MT network cards, respectively.

B. IPsec Packet Size Overhead

The size of IPsec packets depend on the cryptographic algorithms employed, and whether transport or tunnel mode is used. The theoretical VoIP bandwidth requirements without IPsec, and for ESP transport or tunnel mode with or without authentication (i.e., AES-SHA1 and AES) are shown in Fig. 1. The values were computed using the relevant IPsec packet sizes and assuming a 50 packets/sec rate (i.e., 20 ms voice packets). As would be expected, the maximum and minimum IPsec bandwidths are for tunnel mode with ESP authentication, and transport mode without authentication respectively.

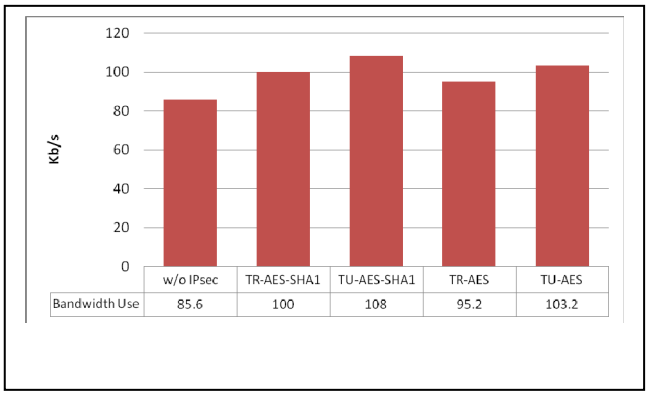

Figure 1. Theoretical VoIP bandwidth with and without IPsec

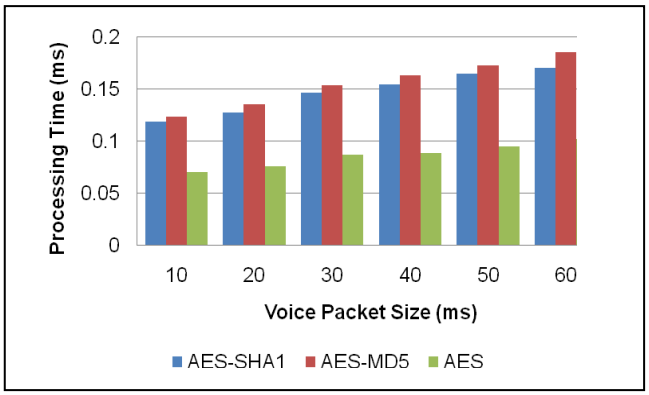

Figure 2. Processing overhead in tunnel mode

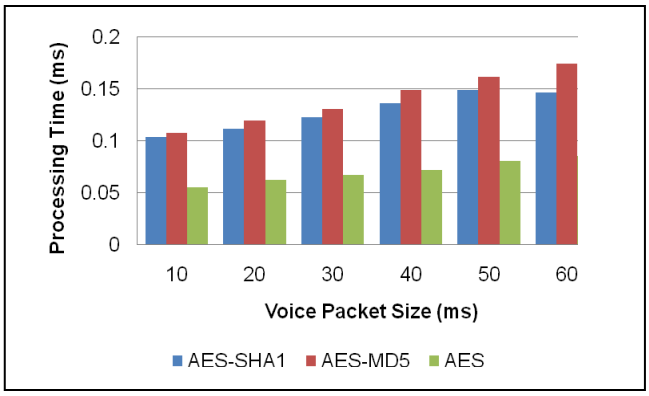

Figure 3. Processing overhead in transport mode

C. IPsec Processing Overhead for Voice

To measure the overhead of IPsec for incoming and outgoing voice traffic, we determined the internal processing times for the corresponding ESP packets. To do this, we added checkpoints to the respective AOs on the bare PCs to capture the processing times.

Figures 2 and 3 show the processing overhead (internal processing times) for incoming ESP voice packets in tunnel and transport modes, respectively. The packet sizes correspond to durations ranging from 10 to 60 milliseconds (ms). ESP configuration included encryption only with AES, encryption with AES and authentication with SHA1 or MD5. With respect to incoming traffic we observed that: 1) for all voice packet

sizes, an MD5 digest has slightly more overhead than a SHA1 digest; 2) the overhead for each cryptographic option increases linearly with increasing the size of voice packets except for a negligible decrease in overhead between 50 and 60 ms packets in transport mode with AES-SHA1; 3) for all cryptographic options and for all packet sizes, the difference between the processing time in tunnel and transport modes is about 0.02 ms.

Figure 4 shows the relative processing overhead increase for incoming traffic compared to outgoing traffic when different packet sizes and different cryptographic options in tunnel and transport modes were used. The overhead increase percentage is largest for ESP with AES only (without authentication) in tunnel mode, and smallest for ESP with either AES-SHA1 or AES-MD5 in transport mode. This is because the overhead of AES decryption and verification for incoming packets is larger than the overhead of AES encryption for outgoing packets, but the additional overhead due to authentication is essentially the same for incoming and outgoing packets of the same size. We also observed that the overhead increase percentage is decreasing with increasing packet size in most cases for the two authentication options in both modes.

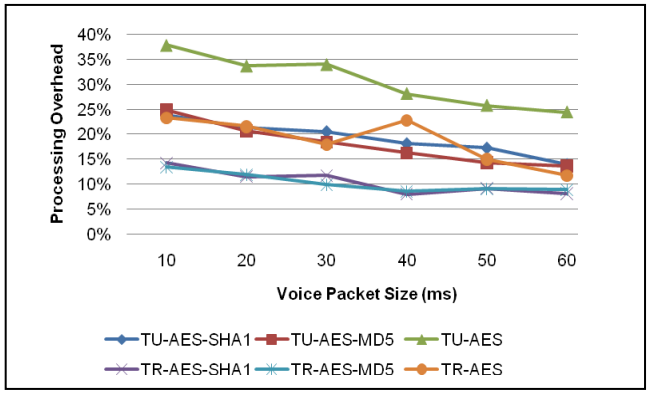

Figure 4. Processing overhead for incoming traffic versus outgoing traffic

D. *Eliminating UDP Overhead*

The UDP header overhead (8 bytes per voice packet) and the associated processing provide port numbers and an optional checksum for voice packets. In reliable secure environments, the checksum will be redundant and elimination of ports may provide additional protection against attacks. This optimization, which requires RTP voice packets to be carried over IP, is easily added to bare PC softphones with IPsec since protocols are intertwined with the application. Figure 5 shows the elimination of UDP processing for voice packets protected with IPsec on a bare PC.

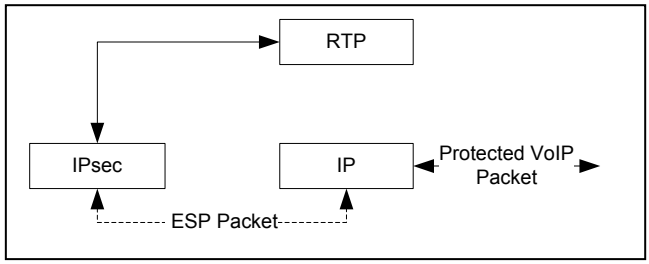

Figure 5. IPsec-secured VoIP without UDP

In Figure 6, the throughput increase percentage with ESP, using AES-SHA1 in tunnel and transport modes, was determined by measuring the actual processing time on a bare PC softphone for incoming and outgoing voice packets without UDP. In general, we observed a 6-14% increase in throughput for incoming packets.

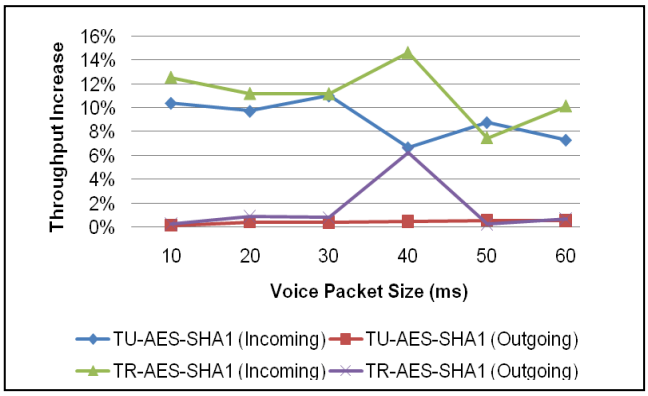

Figure 6. Throughput increase by eliminating UDP

E. *IPsec-Related Inter-Layer Optimizations*

In a conventional OS-based softphone, communication occurs between IP and UDP layers and voice packets are passed between IP and IPsec for ESP processing. In a bare PC softphone, ESP overhead is reduced by directly passing voice packets between the IPsec and UDP handlers for incoming ESP packets. Figure 7 shows that in transport mode the resulting throughput increase for incoming packets due to inter-layer communication in a bare PC softphone, relative to a typical OS-based softphone, is between 8-15%. In this case, it is also seen that the difference in the throughput increase due to varying the voice packet size from 10 ms to 60 ms is less than 3% if 40 ms voice packets are excluded. Further study is needed to determine the reason for the drop in performance with 40 ms packets.

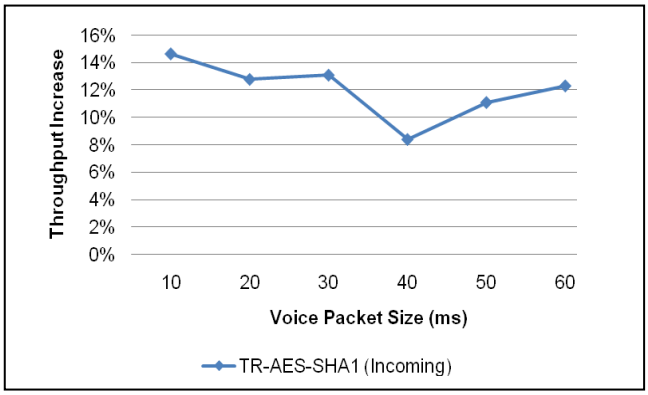

Figure 7. Throughput increase relative to an OS

F. *IPsec Impact on Call Quality*

We measured the values of maximum and mean jitter, and maximum delta (packet inter-arrival time) in transport and tunnel modes using 20 ms voice packets for calls between a pair of bare softphones (BB) and a pair of Linux softphones (LL). The results given in Figures 8 and 9 reflect the call

quality parameters since there was no other traffic on the network. We observed that maximum and mean jitter on a Linux softphone are about 10 and 8 ms, respectively, while the jitter values on a bare PC are negligible (about 0.1 ms). Similarly, maximum delta values are close to the ideal 20 ms on a bare PC, but are almost twice that on a Linux softphone.

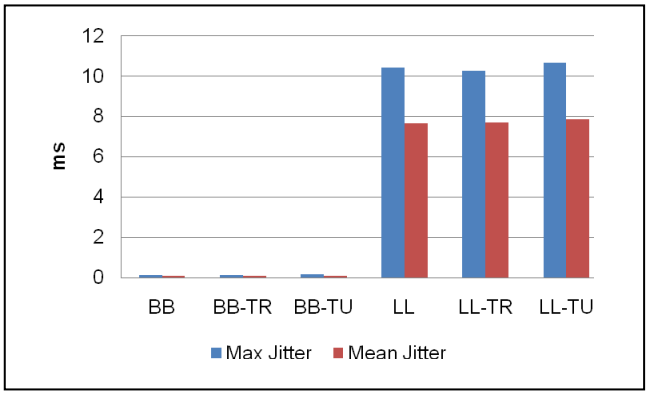

Figure 8. Maximum and Mean Jitter

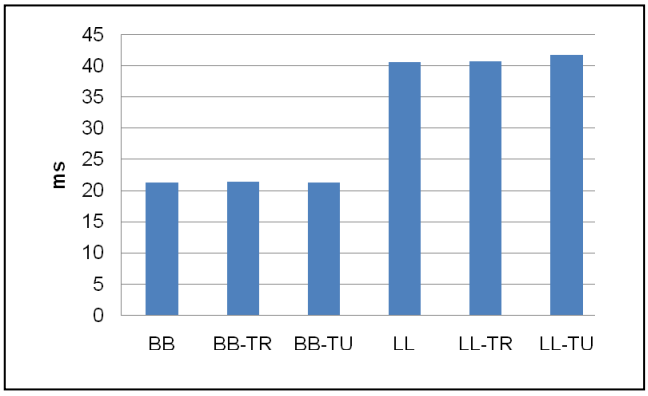

Figure 9. Maximum Delta

IV. CONCLUSION

We evaluated the overhead of IPsec for VoIP using a bare PC softphone. The experimental results using a test LAN show that considering processing overhead, tunnel mode is only slightly more expensive than transport mode. Also, the overhead increase percentage for incoming versus outgoing packets is largest for ESP without authentication in tunnel mode, and smallest for ESP with either authentication option in transport mode. Furthermore, elimination of UDP yields a modest improvement in performance for incoming ESP packets. The throughput increase in a bare PC softphone relative to an OS-based softphone with IPsec is between 8-15%. The minimal values of intrinsic jitter and delta on a bare PC softphone reflect the absence of OS-related overhead when processing ESP voice packets. These results indicate the feasibility of using bare PC softphones with IPsec as a low-overhead option for VoIP in high-security environments.

REFERENCES

[1] R. K. Karne, K. V. Jaganathan, and T. Ahmed, “DOSC: Dispersed Operating System Computing”, OOPSLA ’05, 20th Annual ACM Conference on Object Oriented Programming, Systems, Languages, and Applications, Onward Track, ACM, San Diego, CA, Oct. 2005, pp. 55-62.

[2] R. K. Karne, “Application-Oriented Object Architecture: A Revolutionary Approach,” 6th International Conference, HPC Asia 2002, Bangalore, Karnataka, India, Dec. 2002.

[3] R. K. Karne, K. V. Jaganathan, and T. Ahmed, “How to Run C++ Applications on a Bare PC,” Proceedings of the 6th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, IEEE Computer Society, Washington DC, 2005, pp. 50-55.

[4] S. Kent and K. Seo, "Security Architecture for the Internet Protocol," Dec. 2005, RFC 4301.

[5] S. Kent, "IP Encapsulating Security Payload (ESP)," Dec. 2005, RFC 4303.

[6] E. C. Kaufman, "Internet Key Exchange (IKEv2) Protocol," 2005, RFC 4306.

[7] J. A. Perez, V. Zarate, A. Montes, and C. Gareta, “Quality of service analysis in IPsec VPNs for voice and video traffic,” Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT/IC/W 2006), Feb. 2006, pp. 43-48.

[8] L. Volker, M. Scholler, and M. Zitterbart, "Introducing QoS mechanisms into the IPsec packet processing," 32nd IEEE Conference on Local Computer Networks (LCN 2007), Dublin, Ireland, Oct. 2007, pp. 360-367.

[9] R. Barbieri, D. Bruschi, and E. Rosti, “Voice over IPsec: analysis and solutions,” 18th Annual Computer Security Applications Conference, Dec. 2002, pp. 261-270.

[10] G. H. Khaksari, A. L. Wijesinha, R. K. Karne, L. He, and S. Girumala, “A Peer-to-Peer bare PC VoIP Application,” IEEE Consumer and Communications and Networking Conference (CCNC 2007), Las Vegas, NV, 2007, pp. 803-807.

[11] N. Kazemi, A. L. Wijesinha, and R. Karne, “Design, implementation, and performance of IPsec on a bare PC,” 2nd International Conference on Computer Science and Its Application (CSA 2009), Jeju, S. Korea, Dec. 2009.

[12] (2009), Linphone. [Online]. www.linphone.org.

[13] (2009) WIRESHARK. [Online]. www.wireshark.org.