ARTICLE IN PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computer Networks xxx (2006) xxx–xxx

Computer Networks

www.elsevier.com/locate/comnet

A generic characterization of the overheads imposed by IPsec and associated cryptographic algorithms

Christos Xenakis [*], Nikolaos Laoutaris, Lazaros Merakos, Ioannis Stavrakakis

Communication Networks Laboratory, Department of Informatics and Telecommunications, University of Athens, Athens 15784, Greece

Received 6 May 2005; received in revised form 4 October 2005; accepted 9 December 2005

Responsible Editor: J. Misic

Abstract

This paper presents an assessment of the communication overheads of IPsec and evaluates the feasibility of deploying it on handheld devices for the UMTS architecture. A wide range of different cryptographic algorithms are used in conjunction with IPsec, such as Data Encryption Standard (DES), Advanced Encryption Standard (AES), Message Digest (MD5) and Secure Hash Algorithm 1 (SHA-1). We consider the processing and packetization overheads introduced by these algorithms and quantify their impact in terms of communication quality (added delay for the end-user) and resource consumption (additional bandwidth on the radio interface). We conduct a quantitive analysis based on a detailed simulation model of an IPsec enabled handheld device. We verify our simulation results by comparing against analytic results obtained from an approximate analytic model.
© 2005 Elsevier B.V. All rights reserved.

Keywords: Security performances; Computational overhead; Space overhead; IPsec; DES; AES; MD5; SHA-1; Mobile devices

1. Introduction

Due to the intervention of the open-air medium, the transmission of data over wireless networks generally requires a higher level of security than in wireline networks. The technology of Virtual Private Networks (VPN) [1] has been used for the provision of security services such as confidentiality, integrity and authentication in wireline and, lately, in wireless networks too. A VPN is used for the authentication and the authorization of user access to corporate resources, the establishment of secure tunnels between the communicating parties and the encapsulation and protection of the data transmitted by the network.

VPNs are usually implemented at the application or the network layer of a protocol stack. This paper considers the IPsec suit for deploying VPNs across IP networks. IPsec is a network layer security mechanism that protects traffic on a per connection basis between network layer endpoints, and, thus, is independent from the applications that run above it. IPsec was originally developed for wireline IP

[*] Corresponding author. Tel.: +30 210 7275418; fax: +30 210 7275601.
E-mail addresses: xenakis@di.uoa.gr (C. Xenakis), laoutaris@di.uoa.gr (N. Laoutaris), merakos@di.uoa.gr (L. Merakos), ioannis@di.uoa.gr (I. Stavrakakis).

networks, but it has also been used with wireless IP networks. In these environments, IPsec faces additional challenges introduced by inherently limited resources in the mobile devices and the wireless channels. Such limitations had not been considered in the initial design of the suit. Thus an important open question is "is IPsec a feasible mechanism for implementing VPNs over current and future UMTS networks?" To answer this question, we quantify the *space* (increased bandwidth consumption due to security related additional information on the transmitted packets) and *time* (increased end-to-end delays due to security processing and increased transmission time) overheads of IPsec and project their impact on the current technology. Such a projection will hopefully assist both the mobile users and the network operators in establishing the price of the added VPN functionality through IPsec and in choosing the appropriate suit configuration for their needs.

There is a rather limited literature on the overheads introduced by IPsec [2–4]. A common limitation of existing works is that none of them seems to contain all the currently employed security algorithms and also be independent of specific implementations and platforms. This is precisely the gap that this article intents to fill. Elkeelany et al. [2] look at the processing overhead from employing Data Encryption Standard (DES) (for confidentiality) and Message Digest (MD5) Secure Hash Algorithm 1 (SHA-1) (for authentication security) in conjunction with IPsec. Our work extends this study by jointly considering the Advanced Encryption Standard (AES), which is not considered in [2]. AES is quite important as it is the replacement of DES and 3DES for confidentiality services. Miltchev et al. [3] present a benchmark-based investigation of the performance of IPsec in an OpenBSD system. The same work examines the benefits of using hardware accelerators for speeding up the cryptographic processing. Ganesan et al. [4] perform an experimental evaluation of deploying security algorithms such as, RC4, RC5, MD5 and SHA-1, on low-end embedded systems (Atmel AVR, Mitsubishi M16C, StrongARM, Xscale), as well as on general-purpose systems (SPARC).

Our approach differs fundamentally with all previous ones, in that we avoid making specific assumptions regarding the underlying system, but rather model the induced overheads in a more abstract way. Our analysis is based on the structure and the functionality of the various algorithms and protocols of IPsec. We quantify the number of basic processing operations required by each algorithm, as well as the extra security fields introduced. The peculiarities of the underlying system technology (OS, processor, etc) are left out, since they change rapidly with the shift of technology in both hardware and software. Overall, our approach allows for an effective and simple comparison of different configurations and algorithms of IPsec, independently of specific implementation and platforms. This bridges the gap between purely mathematical analysis and experimental evaluations that target specific implementations and assists in achieving the following twofold objective: (a) provide simulation and analytic frameworks for assessing the processing and space overheads introduced by IPsec; (b) identify possible performance bottlenecks under some typical deployment scenarios of handheld devices and the UMTS radio interface protocol architecture.

The rest of this article is organized as follows. Section 2 presents an overview of IPsec and a quick reference to the most prominent ciphering algorithms used in conjunction with it. Sections 3 and 4 analyze and quantify the processing and space overheads, respectively, of the confidentiality and authentication schemes employed by IPsec. Section 5 presents the simulation model of an IPsec-equipped UMTS mobile device and the obtained simulation results. Section 6 describes the developed analytic model. Section 7 concludes the article.

2. An overview of IPsec

IPsec [5] is a developing standard for providing security at the network layer of the Internet. It facilitates the authentication of the communicating entities, allows them to set up secure IP channels for data exchange, and provides a framework for the employment of different cryptographic algorithms depending on the level of security required by the users and their applications.

IPsec provides two choices of security service through two distinct security protocols: the Authentication Header (AH) protocol [6], and the Encapsulating Security Payload (ESP) protocol [7]. The AH protocol provides support for connectionless integrity, data origin authentication and protection against replays, but it does not support confidentiality. The ESP protocol supports confidentiality, connectionless integrity, anti-replay protection and optional data origin authentication. Both AH and ESP support two modes of operation: transport and tunnel. The transport mode of operation provides end-to-end protection between the communicating end-points by encrypting the IP packet payload. The tunnel mode encrypts the entire IP packet (both IP header and payload) and encapsulates the encrypted original IP packet in the payload of a new IP packet. This fact guarantees that no part of the initial IP packet is exposed to potential threats as the new IP packet is transmitted through intermediate nodes of the network.

IPsec provides an open framework for incorporating a wide range of different cryptographic algorithms for the actual cryptographic task of transforming the original plaintext messages into the transmitted ciphertext. Most ciphering algorithms that are used in conjunction with IPsec are iterative block ciphers. They break the original user data packets into basic blocks of constant size, which are then encrypted independently through a number of encryption rounds. The characteristics of such ciphers depend on the choice of block size, key size and number of rounds. Larger block sizes lead to greater security, but on the other hand reduce the encryption/decryption speed [8]. Similarly, larger keys lead to greater security, but also decrease the encryption/decryption speed. A ‘‘number-of-rounds’’ parameter is usually used for specifying the number of repetitions of the basic encryption process on each basic block of data. In the remaining part of this section, the most prominent ciphering algorithms used in conjunction with IPsec are briefly presented.

The Data Encryption Standard (DES) algorithm [13] is a symmetric (shared secret key) block cipher with block and key size of 64 bits (8 of the 64 bits of the key are used for odd parity, reducing the effective key length). Although widely used, DES has been compromised on several occasions in the past; in fact there exists specialized hardware for breaking it in a few hours [9]. This has lead to the introduction of triple DES (3DES), which is no more than a triple repetition of the basic DES encryption: first the data block is DES-encrypted using an initial key, then the encrypted block is decrypted using a second (different) key and then the new block is re-encrypted using the initial key. This process is equivalent to using a larger effective key length of 112 bits. The obvious disadvantage of 3DES is that it runs three times slower than DES on the same platform.

The Rijndael algorithm, selected as the algorithm of choice for the new Advanced Encryption Standard (AES), [9,10,18], is one of the newest additions to IPsec. Rijndael is a symmetric block cipher that supports different key and block sizes (128, 192, or 256 bits). The AES standardized version of Rijndael, however, is tied to a fixed block size of 128 bits. The initial block is passed through a round transformation function, which is repeated 10 times (respectively, 12 or 14) under a key length of 128 bits (respectively, 192 or 256). Rijndael combines an increased resistance against attacks with an implementation simplicity and, thus, high execution rate. It has proved to be quite durable against differential, truncated differential, linear, interpolation, and Square attacks, [9,11]. Rijndael is quite versatile as it may also serve as a Message Authentication Code (MAC) algorithm, as a hash function and as a pseudo random number generator.

The Message Digest (MD5) [14] and Secure Hash Algorithm 1 (SHA-1), [15], implement so called ‘‘one-way hash functions’’ and are usually used in conjunction with the above cryptographic algorithms for performing authentication. Both of them process input text blocks of 512 bits to generate 128-bit and 160-bit hash values, respectively, for the verification of the correct message transfer. Both apply padding to make the plaintext a multiple of 512 bits, but they cannot be directly used as MAC algorithms, as they do not include a secret key. For that reason, they are used in conjunction with keyed-Hashing for Message Authentication (HMAC), [16,17]. HMAC is a secret key authentication algorithm that provides a framework for incorporating various hashing functions. The combined HMAC-MD5 and HMAC-SHA-1 mechanisms are in position to offer data origin authentication and integrity protection services suitable for IPsec.

In the following two sections, the processing and space overheads introduced by the above algorithms in the framework of IPsec are examined, quantifying the impact of security on the underlying infrastructure when applying IPsec. These derivations are used later to assess the feasibility of IPsec deployment on mobile devices and networks, which are characterized by limited processing power and bandwidth, respectively.

3. Quantification of the *processing overhead* of the confidentiality and authentication schemes employed by IPsec

The IPsec suite entails significant processing work for the encryption and decryption of user data (both of which are applied in a per-packet fashion) to assure the confidentiality and the authentication of the exchanged information (through the computation of integrity-check values (ICV)). Handheld devices have still to operate under rather tight processing and energy constraints, (despite the continuous advances in CPU technology) as compared to the fixed computing devices, for which the IPsec suite was originally designed. Therefore, we first need to quantify the processing overheads imposed by the various protocols employed by the IPsec, in order to assess the feasibility of its deployment on mobile devices. To this end, all processing requirements will be expressed in terms of number of CPU cycles, to facilitate a comparative performance evaluation independently of specific implementations and across different algorithms. The DES, 3DES and AES (for confidentiality) and the HMAC-MD5 and HMAC-SHA-1 (for authentication services) schemes will be investigated. Table 1 summarizes the notation used in the analysis that follows.

3.1. DES, HMAC-MD5 and HMAC-SHA-1

The DES cipher uses a key of 56 bits, and a block of 64 bits. Since DES is a Feistel cipher, it requires the same amount of processing for both encryption and decryption. 3DES results from a triple execution of DES and, thus, requires three times more processing. Let T_{DES} and T_{3DES} denote the number of operations required for encrypting one block of user data with DES and 3DES respectively. The analysis of the two ciphers that appears in [2] has shown that $T_{\mathrm{DES}} = 2697$ and $T_{\mathrm{3DES}} = 8091$. Let S_{d} denote the size of an unencrypted user data packet and let $U_{\mathrm{DES}}(S_{\mathrm{d}})$ and $U_{\mathrm{3DES}}(S_{\mathrm{d}})$ denote the corresponding numbers of operations required to encrypt it with DES and 3DES. Then clearly,

$$U_{\mathrm{DES}}(S_{\mathrm{d}}) = \left\lceil \frac{8 \times S_{\mathrm{d}}}{64} \right\rceil \times T_{\mathrm{DES}}, \tag{1}$$

$$U_{\mathrm{3DES}}(S_{\mathrm{d}}) = \left\lceil \frac{8 \times S_{\mathrm{d}}}{64} \right\rceil \times T_{\mathrm{3DES}} \tag{2}$$

($\lceil\,\rceil$ denotes the ceil function). Consider now a processor that can perform C_{P} Millions Instruction Per Second (MIPS), and let $t_{\mathrm{DES}}(S_{\mathrm{d}}, C_{\mathrm{P}})$ and $t_{\mathrm{3DES}}(S_{\mathrm{d}}, C_{\mathrm{P}})$ denote the time required by this processor for encrypting one user packet of length S_{d} with DES and 3DES respectively. Then,

$$t_{\mathrm{DES}}(S_{\mathrm{d}}, C_{\mathrm{P}}) = \left\lceil \frac{8 \times S_{\mathrm{d}}}{64} \right\rceil \times \frac{T_{\mathrm{DES}}}{C_{\mathrm{P}}}, \tag{3}$$

$$t_{\mathrm{3DES}}(S_{\mathrm{d}}, C_{\mathrm{P}}) = \left\lceil \frac{8 \times S_{\mathrm{d}}}{64} \right\rceil \times \frac{T_{\mathrm{3DES}}}{C_{\mathrm{P}}}. \tag{4}$$

Two common MAC algorithms used in the IPsec suite are the combined HMAC-MD5 and HMAC-SHA-1, which have similar functionality. In both algorithms the hash computation and the hash ver-

Table 1
Notation definition

Symbol	Description
C_{P}	Processor speed in Millions Instructions Per Second (MIPS)
K	Size of the extra appended inner form of the key in MD5 and SHA-1 (512 bits)
Key	Size of a secret key shared by a sender and a receiver (bits)
$K_{\mathrm{i}}, K_{\mathrm{o}}$	Extended forms of the input *Key* (512-bit)
n_k	Number of input blocks for the inner MD5 and SHA-1 algorithm
N_b, N_k, N_r	Parameters of AES that are related to the block size, key length and number of iterations respectively
S_{d}	Size of user data packet (bytes)
s_{p}	Size of the padding field used in MD5 and SHA-1 (bits)
s_{s}	Size of the field that presents the message length in MD5 and SHA-1 (bits)
$T_{\mathrm{HMAC\text{-}MD5}}(n_k)$, $T_{\mathrm{HMAC\text{-}SHA\text{-}1}}(n_k)$	Total number of operations required for applying HMAC-MD5 and HMAC-SHA-1, respectively, as a function of the number of input blocks
$T_{\mathrm{DES}}, T_{\mathrm{3DES}}, T_{\mathrm{Rij}}, T_{\mathrm{MD5}}, T_{\mathrm{SHA\text{-}1}}$	Total number of operations per block required for applying DES, 3DES, AES, MD5 and SHA-1, respectively
$T_{\mathrm{a}}, T_{\mathrm{o}}, T_{\mathrm{s}}$	Number of operations required by a processor for applying a basic byte-wise AND, OR and SHIFT respectively
$t_{\mathrm{DES}}(S_{\mathrm{d}}, C_{\mathrm{P}})$, $t_{\mathrm{3DES}}(S_{\mathrm{d}}, C_{\mathrm{P}})$, $t_{\mathrm{AES}}(S_{\mathrm{d}}, C_{\mathrm{P}})$	Time required by a processor for applying DES, 3DES and AES processing as a function of the user packet size and the processor speed
$U_{\mathrm{DES}}(S_{\mathrm{d}})$, $U_{\mathrm{3DES}}(S_{\mathrm{d}})$, $U_{\mathrm{AES}}(S_{\mathrm{d}})$, $U_{\mathrm{HMAC\text{-}MD5}}(S_{\mathrm{d}})$, $U_{\mathrm{HMAC\text{-}SHA\text{-}1}}(S_{\mathrm{d}})$	Total number of operations required by a processor for applying DES, 3DES, AES, HMAC-MD5 and HMAC-SHA-1 processing as a function of the user packet size

ification are equivalent procedures and, thus, require the same amount of processing. The first step in both the MD5 and SHA-1 algorithms is to pad the original message and make its size a multiple of 512 bits, where the last 64 bits of the last block indicate the length of the message. Subsequently, the algorithms produce a 128-bit and a 160-bits hash values, respectively, of which only a truncated 96-bit portion is used by IPsec. The total number of operations required for MD5 processing per block (512 bits), T_{MD5}, is 720 plus 24 operations for initialization and termination, while for SHA-1 processing, $T_{\mathrm{SHA\text{-}1}}$, is 900 plus 210 operations for initialization and termination [2].

The combined HMAC-MD5 and HMAC-SHA-1 algorithms are formulated as follows:

$$\mathrm{MD5}(K_{\mathrm{o}}, \mathrm{MD5}(K_{\mathrm{i}}, \mathit{Text})),$$
$$\text{SHA-1}(K_{\mathrm{o}}, \text{SHA-1}(K_{\mathrm{i}}, \mathit{Text})),$$

where

$$K_{\mathrm{i}} = \mathit{Key} \oplus \mathit{ipad},$$
$$K_{\mathrm{o}} = \mathit{Key} \oplus \mathit{opad},$$

K_{i} and K_{o} are two extended forms (512-bit) of the input *Key,* which are generated by "exclusive oring" the *Key* with *ipad* (the inner padding (512 bits)) and *opad* (the outer padding (512 bits)) respectively. *Key* is an arbitrary size secret key shared by a sender and a receiver, and $\oplus$ denotes the XOR operation.

For a user packet of size S_{d} bytes, the number of input blocks for the inner MD5 and SHA-1, n_k, is

$$n_k = \left(\frac{8 \times S_{\mathrm{d}} + s_{\mathrm{p}} + s_{\mathrm{s}} + K}{512}\right),$$

where s_{p} is the size (in bits) of the padding field, s_{s} is the size (in bits) of the field that specifies the message length, and K is the size (in bits) of the extra appended inner form of the key.

In the outer MD5 and SHA-1, the output of the inner MD5 (128-bit digest) and SHA-1 (160-bit digest), respectively, are appended to K_{o} and, then, are padded to two 512-bit blocks. Thus, the total number of operations in applying the combined HMAC-MD5 and HMAC-SHA-1, $T_{\text{HMAC-MD5}}(n_k)$, $U_{\text{HMAC-MD5}}(S_{\mathrm{d}})$, $T_{\text{HMAC-SHA-1}}(n_k)$, and, $U_{\text{HMAC-SHA-1}}(S_{\mathrm{d}})$ as a function of the number of input blocks n_k, and the user packet size S_{d}, are

$$T_{\text{HMAC-MD5}}(n_k) = 32 + (2 + n_k) \times 744, \tag{5}$$

$$U_{\text{HMAC-MD5}}(S_{\mathrm{d}}) = 2264 + 744 \times \left(\left\lceil \frac{(8 \times S_{\mathrm{d}}) + 64}{512} \right\rceil\right), \tag{6}$$

$$T_{\text{HMAC-SHA-1}}(n_k) = 32 + (2 + n_k) \times 1110, \tag{7}$$

$$U_{\text{HMAC-SHA-1}}(S_{\mathrm{d}}) = 3362 + 1110 \times \left(\left\lceil \frac{(8 \times S_{\mathrm{d}}) + 64}{512} \right\rceil\right). \tag{8}$$

The factor 32 in Eqs. (5) and (7) derive from the XOR operations performed to produce the inner and the outer keys, K_{i} and K_{o}. Specifically, it results from the division of the size of XOR operands (512 bits) by the word length supported by the processor (i.e. it is assumed to be 32 bits). The outcome of the division (i.e., 16) is multiplied by 2, as the XOR operation occurs twice (one for K_{i} and one for K_{o}). Finally, the required authentication and verification time for HMAC-MD5, $t_{\text{HMAC-MD5}}(n_k, C_{\mathrm{P}})$, and HMAC-SHA-1, $t_{\text{HMAC-SHA-1}}(n_k, C_{\mathrm{P}})$, as a function of the number of input blocks and the processor speed, are

$$t_{\text{HMAC-MD5}}(n_k, C_{\mathrm{P}}) = \frac{32 + (2 + n_k) \times 744}{C_{\mathrm{P}}}, \tag{9}$$

$$t_{\text{HMAC-SHA-1}}(n_k, C_{\mathrm{P}}) = \frac{32 + (2 + n_k) \times 1110}{C_{\mathrm{P}}}. \tag{10}$$

3.2. AES

Rijndael is an iterated block cipher with a block of length $4N_b$ bytes (or N_b (32-bit) words) and a variable key of length $4N_k$ bytes. The encryption of each block of the data involves the following: (a) an initialization phase; (b) $N_r - 1$ iterations of the basic encryption processing of the algorithm; (c) a finalization phase. The version of the Rijndael algorithm that was integrated as part of the AES encryption standard [18] uses a block of 128 bits (i.e., $N_b = 4$) and a key of 128, 192, or 256 bits (i.e., $N_k = 4$, 6, or 8). Depending on the selected key length, the AES standard defines the number of rounds for phase (b) as follows: $N_r(128) = 10$, $N_r(196) = 12$, $N_r(256) = 14$. In [19] the authors have analyzed the Rijndael encryption and have derived simple expressions for T_{Rij}, the computational effort required for encrypting one block of data with this particular cipher. They have expressed this computational effort as a function of the block size, the key size, and the number of processing cycles required for performing basic operations such as a byte-wise AND (T_{a}), a byte-wise OR (T_{o}), and a byte-wise shift (T_{s}). The resulting general expression is

$$T_{\text{Rij-ENC}} = (46N_bN_r - 30N_b)T_{\text{a}} + [31N_bN_r + 12(N_r - 1) - 20N_b]T_{\text{o}} + [64N_bN_r + 96(N_r - 1) - 61N_b)]T_{\text{s}}.$$

By assuming that each basic operation requires one processing cycle, i.e., $T_{\text{a}} = T_{\text{o}} = T_{\text{s}} = 1$, we can derive the corresponding number of processing cycles required for encrypting one block of data with each one of the three standardized flavours of AES (for different key lengths):

$$T_{\text{AES-ENC}}(128) = 6168,$$
$$T_{\text{AES-ENC}}(192) = 7512,$$
$$T_{\text{AES-ENC}}(256) = 8856.$$

Using the same definitions and notation as with DES and 3DES, we can write

$$U_{\text{AES-ENC}}(S_{\text{d}}) = \left\lceil \frac{8 \times S_{\text{d}}}{128} \right\rceil \times T_{\text{AES-ENC}} \quad (11)$$

and,

$$t_{\text{AES-ENC}}(S_{\text{d}}, C_{\text{P}}) = \left\lceil \frac{8 \times S_{\text{d}}}{128} \right\rceil \times \frac{T_{\text{AES-ENC}}}{C_{\text{P}}}, \quad (12)$$

where $T_{\text{AES-ENC}}$ can take either of the following values $T_{\text{AES-ENC}}$ (128), $T_{\text{AES-ENC}}$ (196), or $T_{\text{AES-ENC}}$ (256), depending on the selected key length.

An important difference of Rijndael as compared to other ciphers such as DES and 3DES, is that Rijndael has a non-Feistel structure, meaning that the decryption process makes use of partially different code, which allows for only partial re-use of the encoding circuitry that implements the cipher. The implementation differences are identified in phase (b) of the decryption code, and in particular in the InvMixColumns operation, which uses a different polynomial structure as compared to the corresponding MixColumns operation of the encryption code and, thus, leads to an increased complexity for the decryption. By using the analysis presented in [19], we can obtain an expression for the number of processing cycles for decrypting one block of data.

$$T_{\text{Rij-DEC}} = T_{\text{Rij}} + 96N_bT_{\text{a}} + (N_r - 1) \times (72N_bT_{\text{o}} - 32N_bT_{\text{s}}). \quad (13)$$

The above expression (Eq. (13)) points to the fact that the Rijndael decryption is computationally more expensive than the encryption (the actual difference being $96N_bT_{\text{a}} + 72N_bT_{\text{o}} - 32N_bT_{\text{s}}$ operations for each of the $N_r - 1$ rounds of phase (b)). Using this expression, we can obtain the corresponding number of processing cycles required for decrypting one block of data with each one of the three standardized flavours of AES (for different key lengths):

$$T_{\text{AES-DEC}}(128) = 10992,$$
$$T_{\text{AES-DEC}}(192) = 13408, \quad (14)$$
$$T_{\text{AES-DEC}}(256) = 15824.$$

From these values, we can obtain $U_{\text{AES-DEC}}(S_{\text{d}})$ and $t_{\text{AES-DEC}}(S_{\text{d}})$ similarly to the encryption case. Notice that the number of operations required for the decryption is significantly higher than for the encryption. Thus, there have been some efforts for reducing this asymmetry through faster implementations (see for example [19–21]).

4. Quantification of the *space overhead* of the confidentiality and authentication schemes employed by IPsec

Both ciphering and the IPsec packetization increase the final size of the transmitted packets, thereby creating *space overhead*. The ciphering overhead is created by padding the original packet in order to make its size a multiple of the basic block size of the ciphering algorithm. The IPsec packetization overhead is created by the additional IPsec-specific fields that are added to the protected IP packets.

The IPsec packetization overhead depends on the selected security protocol. Fig. 1(a) presents the format of an ESP protocol packet. The ESP header, which includes the security parameters index and the sequence number fields, is inserted into the IP packet immediately prior to the transport-layer header. The ESP trailer, which contains the padding, the pad length, and the next header fields, is placed after the IP packet. When the authentication service of IPsec is selected, an additional field—the ESP authentication data field—is added after the ESP trailer. Therefore, the space overhead that corresponds to the ESP security protocol consists of: (i) the fixed-size fields (10 bytes), (ii) the optional authentication data field (12 bytes), and (iii) the variable length padding applied (that ranges from 0 to 8 bytes). Fig. 1(b) presents the format of the AH protocol header. The header consists of the fixed-size fields (i.e., Next header, payload length, reserved, security parameter index, and sequence number) (12 bytes) and the field of authentication data (12 bytes).

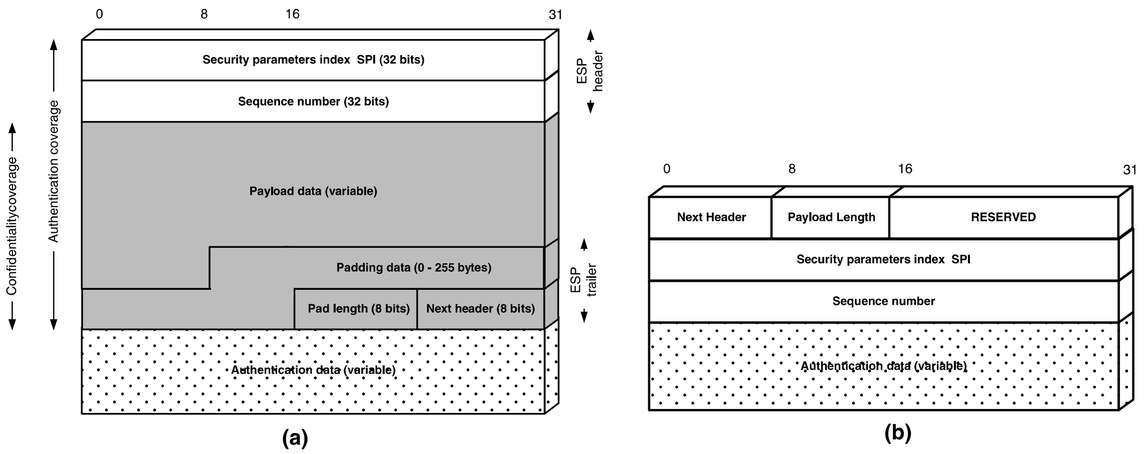


Fig. 1. The format of (a) an ESP protocol packet and (b) the AH protocol header.

The operation mode of IPsec and the supported security services (authentication, confidentiality, or combined) are also contributing to the space overhead. In transport mode the aforementioned IPsec-specific fields (i.e., ESP header, AH header, ESP trailer and authentication data field) are inserted within the transmitted IP packet as shown in Fig. 2(a). Thus, the total space overhead per-packet for the AH protocol is 24 bytes, and for the ESP protocol is 10 bytes plus the variable length of the padding field for confidentiality service, and 22 bytes plus the variable length of the padding field for combined confidentiality and authentication services. On the other hand, in tunnel mode, a new IP header is added in each IPsec-protected packet in addition to the IPsec-specific fields (Fig. 2(b)), which leads to a higher space overhead than in transport mode. The total space overhead is thus given by the sum of the above values (transport mode) and the size of the new IP header (20 bytes).

The previous discussion allows us to express the space overhead of IPsec as a function of the user packet size S_d, and elaborate specific formulas. The formulas present the size of protected packet for both security protocols (ESP and AH) that are configured to provide confidentiality (CNF), authentication (ATH), and combined confidentiality and authentication (CNF-ATH) security services in both modes of IPsec operation. Let $SP_{X-Y}(S_d)$, and $SL_{X-Y}(S_d)$, denote the size of protected packets using the protocol X (ESP or AH) that provides Y (CNF, ATH, or CNF-ATH) security services. The employed security algorithms are selected from the set of the analyzed ciphers (i.e., DES, 3DES AES, HMAC-MD5 and HMAC-SHA-1). Table 2 explains the notation used in the formulas, and Table 3 presents the details of the formulas.

The next section of this paper presents a simulation study that attempts to assess the feasibility of IPsec deployment on mobile devices and networks. The simulation model incorporates the analyzed

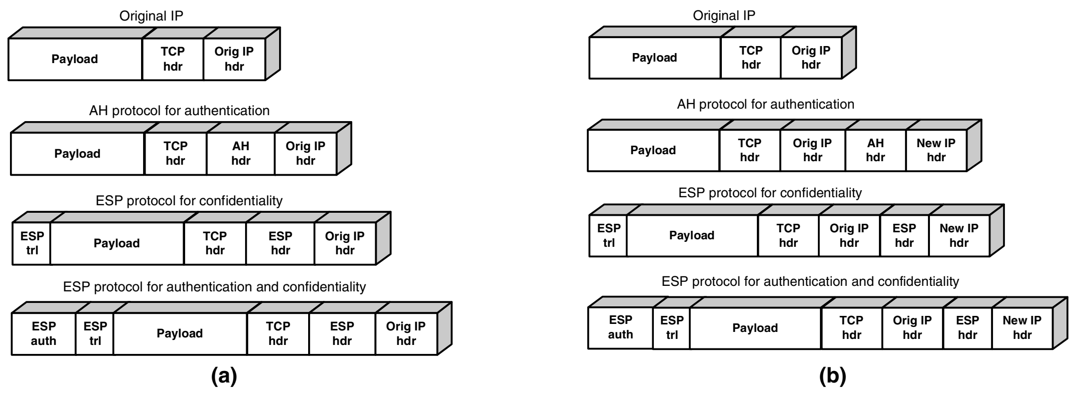

Original IP
Payload
TCP hdr
Orig IP hdr
AH protocol for authentication
AH hdr
ESP protocol for confidentiality
ESP trl
ESP hdr
ESP protocol for authentication and confidentiality
ESP auth
(a)
New IP hdr
(b)

Fig. 2. The packet format of AH and ESP protocol in (a) transport and (b) tunnel mode of operation.

Table 2
Notation definition

Symbol	Description
AuT_{ESP}	Size of the authentication data field of the ESP protocol (12 bytes)
Bl	Block size of the encryption algorithms (8 bytes for DES, and 16 bytes for AES)
H_{ESP}, H_{IP}, H_{TCP}	Size of the ESP header (8 bytes), the IP header (20 bytes), and the TCP header (20 bytes), respectively
$RP(S_{\mathrm{d}})$, $RL(S_{\mathrm{d}})$	Ratio of the actual payload to the total packet length, as a function of user packet size for transport and tunnel mode of operation respectively
S_{d}	User packet size (bytes)
$SP(S_{\mathrm{d}})$, $SL(S_{\mathrm{d}})$	Size of IPsec-protected packets as a function of user packet size for transport and tunnel mode of operation respectively (bytes)
Tr_{ESP}	Size of the fixed part of the ESP trailer (2 bytes)

Table 3
Formulas that give the size of protected packets as a function of user packet size for different configurations of IPsec

Notation	Description formula	Final formula[a]
$SP_{\mathrm{ESP\text{-}CNF}}(S_{\mathrm{d}})$	$\left\lceil \frac{S_{\mathrm{d}}+H_{\mathrm{TCP}}+Tr_{\mathrm{ESP}}}{Bl} \right\rceil \times Bl + H_{\mathrm{ESP}} + H_{\mathrm{IP}}$	$\left\lceil \frac{S_{\mathrm{d}}+22}{Bl} \right\rceil \times Bl + 28$
$SL_{\mathrm{ESP\text{-}CNF}}(S_{\mathrm{d}})$	$\left\lceil \frac{S_{\mathrm{d}}+H_{\mathrm{TCP}}+H_{\mathrm{IP}}+Tr_{\mathrm{ESP}}}{Bl} \right\rceil \times Bl + H_{\mathrm{ESP}} + H_{\mathrm{IP}}$	$\left\lceil \frac{S_{\mathrm{d}}+42}{Bl} \right\rceil \times Bl + 28$
$SP_{\mathrm{ESP\text{-}ATH}}(S_{\mathrm{d}})$	$\left\lceil \frac{S_{\mathrm{d}}+H_{\mathrm{TCP}}+Tr_{\mathrm{ESP}}}{4} \right\rceil \times 4 + H_{\mathrm{ESP}} + H_{\mathrm{IP}} + AuT_{\mathrm{ESP}}$	$\left\lceil \frac{S_{\mathrm{d}}+22}{4} \right\rceil \times 4 + 40$
$SL_{\mathrm{ESP\text{-}ATH}}(S_{\mathrm{d}})$	$\left\lceil \frac{S_{\mathrm{d}}+H_{\mathrm{TCP}}+H_{\mathrm{IP}}+Tr_{\mathrm{ESP}}}{4} \right\rceil \times 4 + H_{\mathrm{ESP}} + H_{\mathrm{IP}} + AuT_{\mathrm{ESP}}$	$\left\lceil \frac{S_{\mathrm{d}}+42}{4} \right\rceil \times 4 + 40$
$SP_{\mathrm{ESP\text{-}CNF\text{-}ATH}}(S_{\mathrm{d}})$	$\left\lceil \frac{S_{\mathrm{d}}+H_{\mathrm{TCP}}+Tr_{\mathrm{ESP}}}{Bl} \right\rceil \times Bl + H_{\mathrm{ESP}} + H_{\mathrm{IP}} + AuT_{\mathrm{ESP}}$	$\left\lceil \frac{S_{\mathrm{d}}+22}{Bl} \right\rceil \times Bl + 40$
$SL_{\mathrm{ESP\text{-}CNF\text{-}ATH}}(S_{\mathrm{d}})$	$\left\lceil \frac{S_{\mathrm{d}}+H_{\mathrm{TCP}}+H_{\mathrm{IP}}+Tr_{\mathrm{ESP}}}{Bl} \right\rceil \times Bl + H_{\mathrm{ESP}} + H_{\mathrm{IP}} + AuT_{\mathrm{ESP}}$	$\left\lceil \frac{S_{\mathrm{d}}+42}{Bl} \right\rceil \times Bl + 40$
$SP_{\mathrm{AH}}(S_{\mathrm{d}})$	$(S_{\mathrm{d}} + H_{\mathrm{TCP}} + H_{\mathrm{IP}} + 24)$	$(S_{\mathrm{d}} + 64)$
$SL_{\mathrm{AH}}(S_{\mathrm{d}})$	$(S_{\mathrm{d}} + H_{\mathrm{TCP}} + 2 \times H_{\mathrm{IP}} + 24)$	$(S_{\mathrm{d}} + 84)$

[a] The final formula is derived after substituting the values of the variables and performing some additional simple algebraic manipulation.

models for processing and space overhead and subjects them to typical data traffic [22].

5. Simulation study

Fig. 3 depicts a block diagram of the IPsec-equipped MS that is considered in the following simulation study. The model consists of the following components: (i) a traffic generator for the creation of non-real time traffic at the network layer according to the parameters that are defined in a next paragraph; (ii) an IPsec/IP processor queue where packets accumulate before entering the processor that applies the cryptographic algorithms and the functionality of IPsec/IP according to the selected mode of operation of IPsec; and (iii) a transmitter module that incorporates the radio interface protocol architecture of UMTS in the user plane.

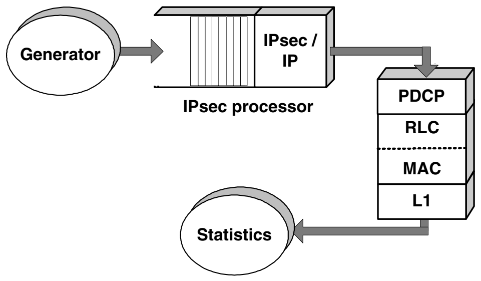


Fig. 3. Model of an IPsec-equipped MS.

The transmitter module is further divided into four layers: the physical layer (L1), the Medium Access Control (MAC), the Radio Link Control (RLC) and the Packet Data Convergence Protocol (PDCP). The physical layer is responsible for the Forward Error Correction (FEC) of transport channels (channel coding), error detection using Cyclic Redundancy Code (CRC) and RF processing. The FEC scheme aims at detecting and recovering from transmission errors by adding information in the packets that can be used for their reconstruction at the receiver. It is implemented by employing a convolutional code that introduces 1/2 code rate, which indicates the space overhead of the coder (the output of the coder is twice as many bits used for input). The channel coding (FEC) is combined with a CRC error detection function that verifies the results of FEC. Using this function, erroneous packets are detected and indicated to higher layers (RLC) for retransmission. The length of the CRC polynomial used is 16 bits [23].

On top of the physical layer the MAC and RLC provide link layer functionality. The MAC layer provides logical channels to the RLC and maps logical channels into transport channels using the non-transparent transmission mode [24]. The RLC layer provides an acknowledged mode of data transfer by employing an Automatic Repeat Request (ARQ) mechanism. This mechanism tries to recover errors by retransmitting flawed packets indicated by the physical layer (error detection function) [25] and thus putting additional load on the radio interface. Finally, the PDCP maps the IP layer to the RLC [26].

The employed simulated traffic represents non-real time user traffic according to the reference model defined by the 3GPP in [22]. It is assumed that there exists an active user that generates packet sessions. Each session involves bursty sequences of packets. The mean user data rate is denoted λ_{data} and ranges from 128 Kbit/s to 1.2 Mbit/s. Packet inter-arrival times between subsequent user packets in a packet call are modeled by an i.i.d. random variable that follows an exponential distribution with parameter μ_{d}. The sizes of user packets are modeled by an i.i.d. random variable S_{d} that follows the truncated Pareto distribution $f_{S_{\text{d}}}(x)$:

$$f_{S_{\text{d}}}(x) = \begin{cases} \frac{ak^a}{x^{a+1}}, & k \leqslant x < \mu, \\ \left(\frac{k}{m}\right)^a, & x = \mu. \end{cases} \tag{15}$$

The parameters k and m define the minimum and the maximum user data packets respectively and the parameter a defines the skewness of the distribution (the default values are $a = 1.1$, $k = 81.5$ bytes and $m = 66\,666$ bytes [22]). The average packet size is $\mu_n = 480$ bytes and the radio channel capacity is 2 Mbps (total rate including all the management and control information). The packet error rate (PER) parameter over the radio interface, which also indicates the percentage of retransmissions at the RLC layer, is 2%. The mobile device is assumed to be equipped with an embedded processor with a processing rate C_{p} in the range of 100–500 Millions of Instructions Per Second (MIPS) [27]. Table 4 summarizes the values of the basic simulation parameters.

A total of fifty-three (53) different security scenarios are considered. They include: (i) the two different modes of operation of IPsec (transport and tunnel); (ii) the two different protocols of IPsec (ESP and AH); (iii) several different cryptographic algorithms that provide different levels of security: no security, pure confidentiality (DES, 3DES and AES with variable key length for the encryption and decryption process), pure authentication (MD5 and SHA-1), and combined confidentiality and authentication (DES + MD5, 3DES + MD5, DES + SHA-1, 3DES + SHA-1, AES(128)Enc + MD5, AES(128)Enc + SHA-1, AES(192)Enc + MD5, etc). The evaluation of the different scenarios is based on the following performance metrics: (i) the system throughput, (ii) the packet latency, and (iii) the data rate increase due to IPsec. In the next paragraphs we summarize our observations from the simulation experiments. The parameters that will be examined in the following simulations include: the offered traffic load, the service rate of the IPsec/IP processor queue, and the characteristics of the radio interface.

Table 4
Simulation parameters setting

Simulation parameters	Base values
Mean data rate λ_{data}	128 Kbit/s–1.2 Mbps
MS processing speed C_{MS}	100–500 MIPS
Average size of datagram μ_n	480 bytes
Radio channel capacity	2 Mbps
Packet error rate (PER)	2%

In the majority of the employed security scenarios, the encryption and decryption are symmetric processes and, thus, they consume the same amount of processing. For that reason we have selected to develop a simulation model that represents only the IPsec encryption, and use it as a basis for the accomplished performance evaluation and the comparison of the different security scenarios. However, as mentioned previously (Section 3.2), the AES cipher has a non-Feistel structure and the decryption is computationally more expensive than encryption. Thus, both AES processes (encryption and decryption) with all the possible key lengths (i.e., 128, 192, 256) are applied to the developed simulation model. This fact facilitates the assessment of the computational difference between the encryption and the decryption process in AES, and the comparison of both AES processes with the rest of the employed ciphering algorithms.

5.1. System throughput

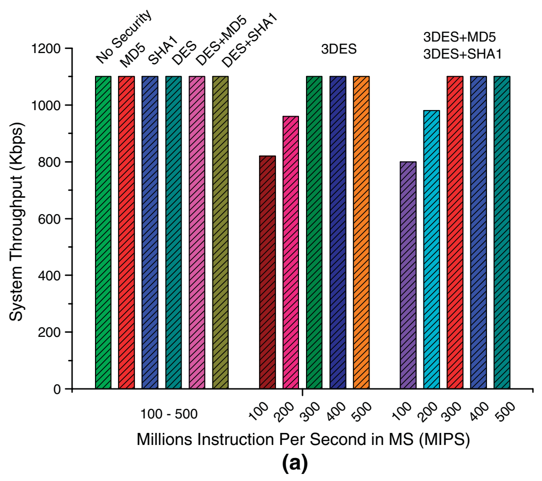
Fig. 4 depicts the system throughput as a function of the processing speed of the MS for the above security scenarios. The IPsec mode of operation has a negligible effect on the system throughput and, thus, the presented diagrams represent both modes of operation. One may observe that the more "lightweight" security schemes like MD5, SHA-1, DES, DES + MD5 and DES + SHA-1 do not degrade the system throughput, as they add a rather limited amount of processing (see Fig. 4(a)). This points to the fact that a processing rate of 100 MIPS and above should be enough for handling the added processing of these lightweight schemes. For the above combinations of security schemes and processing rates, the bottleneck in terms of throughput is dictated by the capacity of the radio channel. More complex security schemes like 3DES, 3DES + MD5 and 3DES + SHA-1 provide for an increased resistance against attacks but pose higher processing requirements and, thus, reduce the system throughput when the MS processing rate is below 300 MIPS (which appears to be the borderline minimum for employing these schemes).

Regarding AES, although it requires less processing than 3DES, it also lowers the system throughput when there is not sufficient processing capability at the MS (see Fig. 4(b)). The throughput of the encryption process of AES presents higher values compared to the decryption. The lightest flavor of AES, i.e., the one that provides encryption with a key length of 128 bits, has almost no effect on the system's throughput. Increasing the key length, however, puts more strain on the processor and this can translate into reduced throughput in cases that the MS processing rate is bellow 300 MIPS (which also appears to be the boarder for employing AES). Combining confidentiality with authentication services by adding MD5 or SHA-1 to AES increases even more the strain on the processor. This extra strain is, however, relatively small as compared to the one imposed by the encryption scheme and thus is hardly visible on the figures.

5.2. Total delay

Except for its impact on the system's throughput, a security scheme increases the total delay for transmitting a user packet. Fig. 5 shows the total delay as a function of the user data rate for the various security schemes and a processing rate of 100 MIPS. Sole application of authentication services, like MD5 and SHA-1, hardly has an impact on the total delay (see Fig. 5(a)). DES encryption presents higher delay values than authentication services, but its application on the system does not considerably influence the data transfer. On the contrary, AES encryption and decryption with a 256 bit key length (labeled AES(256)Enc and AES(256)Dec, respectively, in the corresponding figure) and 3DES have stronger impact on the total delay. Moreover, these scenarios under sufficiently high user data rates lead to excessive delay values, which point to the fact that the user data rate has exceeded the maximum capacity of the MS.

Fig. 5(b) presents the total delay values for the various configurations of the AES algorithm in the framework of IPsec. The simulation results have verified that the decryption process of AES presents



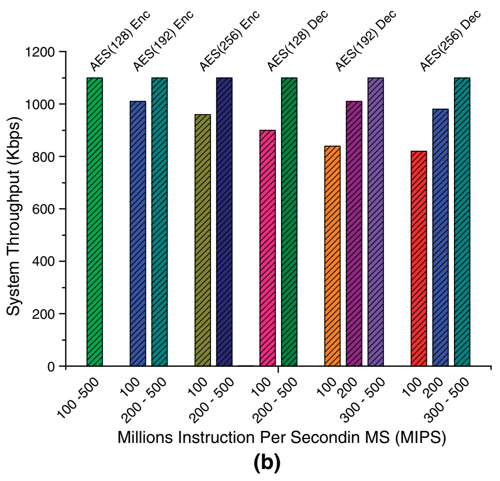


Fig. 4. System throughput as a function of the processing speed of the MS for both modes of IPsec operation, and different security scenarios like (a) no security, MD5, SHA-1, DES, DES + MD5, DES + SHA-1, 3DES, 3DES + MD5 and 3DES + SHA-1, (b) AES with different key size for both encryption and decryption process.

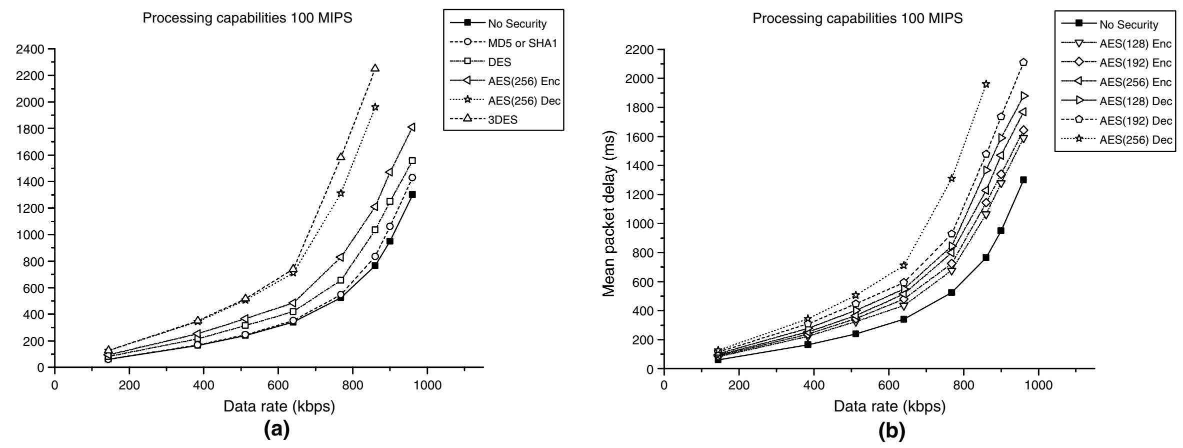


Fig. 5. Mean total delay as a function of mean data rate for 100 MIPS processing rate at the MS and (a) MD5, SHA-1, DES, AES encryption and decryption with a 256 bit key and 3DES (b) AES encryption and decryption with variable key length.

higher delay values than encryption. Moreover, they have verified that as the size of the key used by AES increases, the algorithm execution becomes more complex resulting in a higher delay values for data transfer. Thus, the plotted delay values quantify the processing differences of the different options of AES algorithm in the framework of IPsec in the considered environment.

Fig. 6 presents the total delay of the combined confidentiality and authentication security services using DES + SHA-1 and AES(128)Enc + SHA-1 algorithms, as a function of the user data rate and for a processing rate of 100 MIPS. It compares the above total delay values to the total delay of the pure confidentiality security services using DES and AES(128)Enc algorithms, respectively. Observed that the addition of authentication security services hardly increases the total packet delay values, since authentication represents a relatively lightweight (from the processing overhead point of view) security service. In addition, Fig. 6 shows that DES and AES encryption with 128-bit key have a similar behavior and add marginally to the total delay as compared to the no-security scenario. The source data of Fig. 6 as well as the confidence intervals are presented in the table included in Appendix A of this paper.

For a greater MS processing rate of 200 MIPS, there is a similar qualitative behavior as with the abovementioned 100 MIPS case. However, the absolute delay values become smaller, owing to the shorter time spent on the IPsec/IP processor queue. In fact, with such a processing rate, some of the lightweight security schemes incur a total delay that approaches the one of the no security scenario (see Fig. 7(a)). Increasing the MS processing rate further to 500 MIPS (Fig. 7(b)) pushes the delay curves of the various security schemes very close to the no security curve, which means that in this case IPsec has almost a negligible impact on the system's performance with respect to the delay.

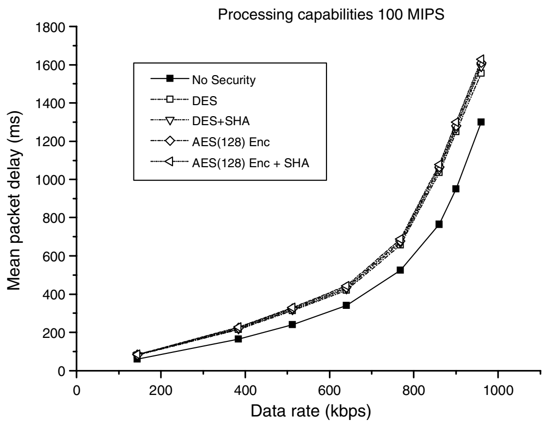


Fig. 6. Mean total delay as a function of mean data rate for 100 MIPS processing rate at the MS and DES, DES + SHA-1, AES(128)Enc and AES(128)Enc + SHA-1.

5.3. Rate increase of the protected data

Apart from the processing overhead due to the IPsec/IP processor queue, a security mechanism adds an additional *space overhead* by increasing the size of the final transmitted user packets. The increase owes to the extra security fields encapsulated with the user data in the final packets, and

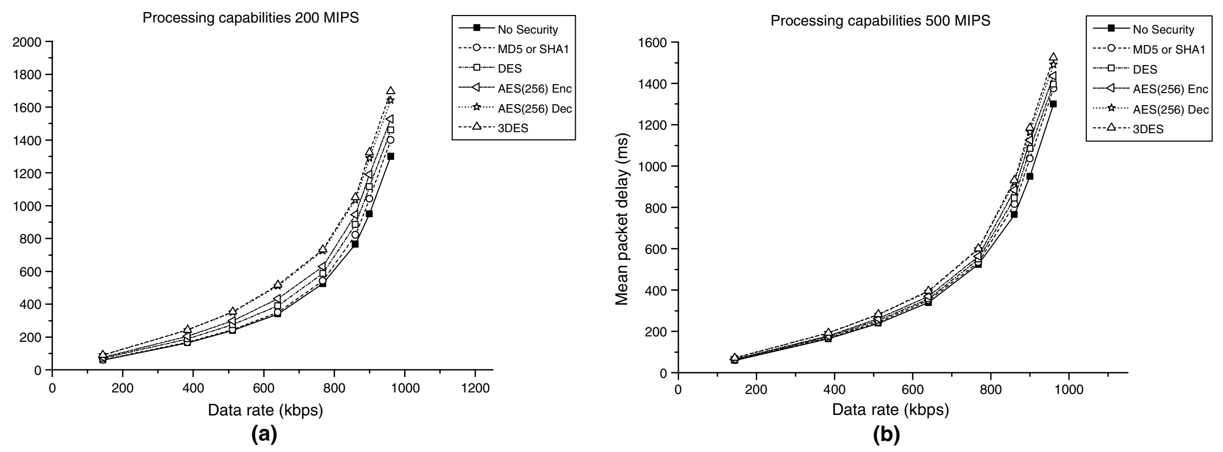


Fig. 7. Mean total delay as a function of mean data rate for MD5, SHA-1, DES, AES(256)Enc, AES(256)Dec and 3DES and (a) 200 (b) 500 MIPS processing rate at the MS.

has the natural effect of increasing both the packet transmission time and the final output rate (thereby consuming more bandwidth on the wireless channel). Fig. 8 depicts the rate increase of the protected data as a function of the actual user data rate. As was expected from the discussion of the space overhead of IPsec (Section 4), the tunnel mode of operation leads to a higher overhead than the transport mode. One may observe that for each mode of operation of IPsec, the confidentiality services (using AES or DES or 3DES) impose the same rate increase; the slight difference between the space overhead of AES and DES (3DES) causes a negligible difference to the rate increase of the protected data. The same applies among the pure authentication, and the combined authentication and confidentiality security services (i.e., MD5, SHA-1, DES + MD5, AES + SHA-1, etc.). The confidentiality security services impose the lowest rate increase of the protected data; while the pure authentication security services and the combined confidentiality and authentication services impose the highest increase to the final output rate.

6. Analytic model of an IPsec-equipped MS

In this section we develop a simple analytic model for the abstract MS that is depicted in Fig. 9. The analysis is carried out by modeling the IPsec processor and the transmitter as a system of two independent M/G/1 queues in tandem. The analysis aims at both verifying the simulation results and providing a faster alternative to them.

6.1. First queue (IPsec processor)

The first queue is an M/G/1 queue with the following characteristics: (i) a Poisson arrival process of rate λ for modeling the arrivals of data packets from the user; (ii) i.i.d. service times X_1 with expected value $E\{X_1\}$ and expected square value $E\{X_1^2\}$. Let μ_1 denote the constant service rate of the server of the first queue (to be defined in detail shortly). Then $E\{X_1\}$ and $E\{X_1^2\}$ can be written as

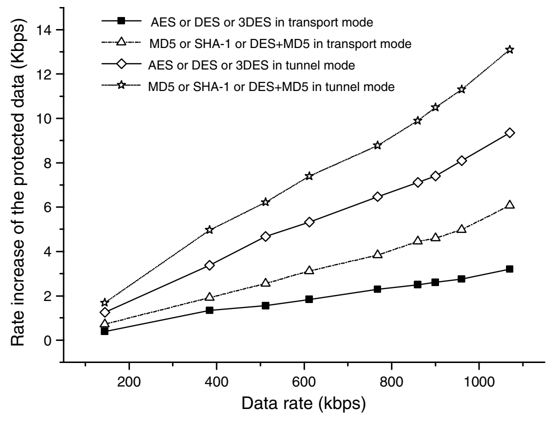


Fig. 8. Rate increase of the protected data as a function of the actual data rate for various security services.

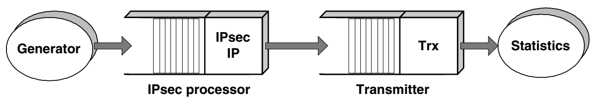


Fig. 9. Analytic model.

functions of μ_1 and $f_{S_d}(x)$, the probability density function of user packets which, as with Section 5, are assumed to be following a truncated Pareto distribution with parameters k, m, a. Thus,

$$E\{X_1\} = \int_k^m \frac{x}{\mu_1} f_{S_d}(x)\,dx = \frac{a(mk^a - km^a)}{m^a\mu_1(1-a)} + m(k/m)^a/\mu_1, \tag{16}$$

$$E\{X_1^2\} = \int_k^m \left(\frac{x}{\mu_1}\right)^2 f_{S_d}(x)\,dx = \frac{a(m^2k^a - k^2m^a)}{m^a\mu_1^2(2-a)} + m^2(k/m)^a/\mu_1^2. \tag{17}$$

Let W_1 denote the mean total delay at the first queue (queuing and transmission components); W_1 is given by the well known Pollaczek–Khinchin (P–K) formula, i.e.,

$$W_1 = E\{X_1\} + \frac{\lambda E\{X_1^2\}}{2(1-\lambda E\{X_1\})}. \tag{18}$$

The service rate μ_1 of the IPsec processor queue depends on: (i) the speed of the processor (C_p) in instructions per second; (ii) the block-size N_X used by the employed cryptographic algorithm X for encrypting user data (iii) the number of instructions required by the cryptographic algorithm X for encrypting one block of user data of size N_X (in the previous section this quantity has been denoted T_X). The exact relationship giving μ_1 is $\mu_1 = (N_X/8)(C_p/T_X)$. An important observation with regard to the expression of $E\{X_1\}$ and $E\{X_1^2\}$ is that a user packet of size x requires the processing of $\lceil x/N_b \rceil$ blocks of data under a block-cipher with block size N_b. In order to simplify the derivation, we have neglected the ceiling function and, thus, the analytic model may account for up to minus one blocks per user packet. As will be shown later, this approximation has a rather minor effect on the accuracy of the model and, thus, is worth performing it in order to simplify the analysis.

6.2. Second queue (transmitter)

The arrival process of the transmitter queue is given by the output process of the IPsec processor queue and, thus, is no longer a Poisson process. We will employ an *independence approximation* and assume it to be Poisson nevertheless. The basis for making this assumption is that under heavy load conditions and highly variable service times at the first queue, the independence approximation can produce usable results in terms of accuracy.[1] The aforementioned conditions hold to a large extent true for our application and, thus, as will be shown in the sequel, the numerical results from our approximate analytic model compare favourably to the simulations results of the actual system. This makes the approximate analytic model a useful tool for conducting a first qualitative analysis of an IPsec-equipped MS without having to resort to laborious simulations. More security schemes (possibly future ones) and different parameter sets can thus be evaluated quickly without needing a simulation study.

Under the above-mentioned approximation, the second queue becomes too an M/G/1 queue with the same Poisson arrival process and i.i.d. service times X_2 that correspond to the time that is required for transmitting the IPsec-protected user packets over the wireless link. To write $E\{X_2\}$ and $E\{X_2^2\}$ we will take into consideration the following facts: (i) the wireless channel has a constant transmission rate of μ_2 bytes per second and (ii) the user data packets have sizes that correspond to a truncated Pareto distribution. For the second queue, however, the truncated Pareto distribution will be a shifted version of the original one (the shifting being on the x-axis), because each transmitted packet has an additional space overhead of R bytes due to the encryption related information inserted by the employed security scheme and the various headers added by the four layers of the transmitter protocol stack (PDCP/RLC/MAC/L1). Thus,

$$E\{X_2\} = \int_k^m \frac{x+R}{\mu_2} f_{S_d}(x)\,dx = \frac{k^a(ma - R + aR) - m^a(ka + R - aR)}{m^a\mu_2(1-a)} + \frac{(m+R)(k/m)^a}{\mu_2}, \tag{19}$$

$$E\{X_2^2\} = \int_k^m \left(\frac{x+R}{\mu_2}\right)^2 f_{S_d}(x)\,dx = \frac{A+B+C}{m^a\mu_2^2(1-a)(a-2)} + \frac{(m+R)^2(k/m)^a}{\mu_2^2}. \tag{20}$$

The parameters A, B and C used in Eq. (20) are as follows:

$$A = -3ak^aR^2 + k^aR^2a^2 + 2k^aR^2 - m^2k^aa + k^aa^2m^2 + 2k^aRa^2m,$$
$$B = -4mk^aRa + 3m^aR^2a - m^aR^2a^2 - 2m^aR^2 + m^aak^2 - m^aa^2k^2,$$
$$C = -2m^aRa^2k + 4m^aRak.$$

[1] Such approximations are known to be leading to a smaller average delay than the actual system of two queues in tandem and this is also the case in our results here [12].

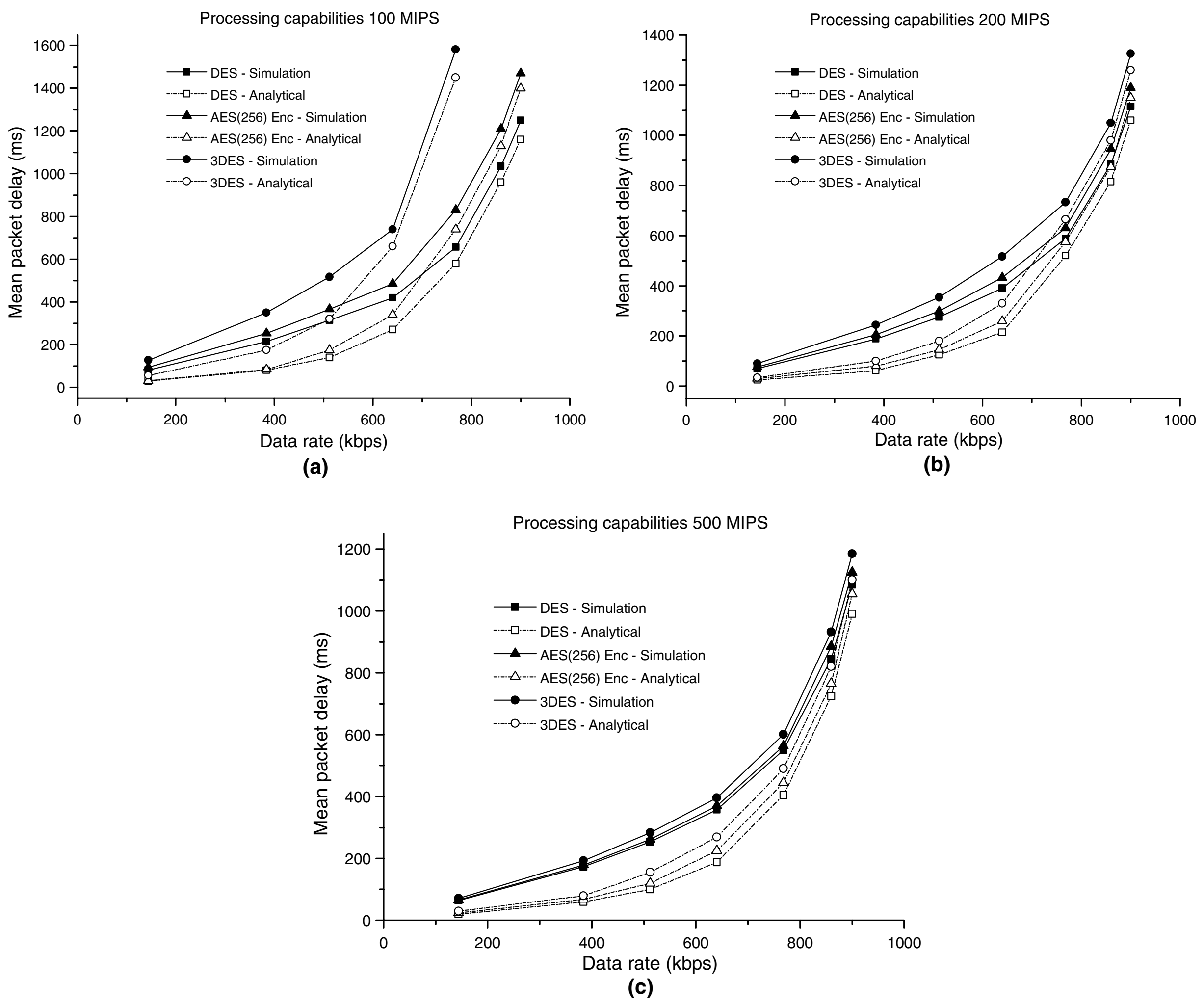


Fig. 10. Total delay obtained from the analytical and the simulation model as a function of the actual data rate for DES, AES(256)Enc and 3DES security scenarios and for (a) 100 MIPS (b) 200 MIPS and (c) 500 MIPS processing rate at the MS.

We can now use the P–K formula in order to compute W_2, the mean total delay at the second queue. In doing so, we take notice to increase the input rate λ by 2% in order to account for packet retransmissions by the RLC layer of the transmitter. The overall delay (encryption and transmission components) is then taken from $W = W_1 + W_2$.

In Fig. 10 we plot the total delay obtained from the above analysis against the one obtained from the simulation experiments of the previous section. We show results for 100, 200 and 500 MIPS for some indicative scenarios such as DES, AES(256)Enc and 3DES. One may easily conclude that the analytic results capture satisfactorily the qualitative behavior in terms of the delay. Observe, however, that the absolute delay values are lower in the analysis than in the simulation. This is in accordance to our expectation and owes to (i) the use of the independence approximation for the arrival process of the second queue; (ii) the disregard of the ceiling function in the computation of the number of encryption blocks that correspond to one user packet of size S_d, and (iii) the disregard of the ceiling function in the computation of the space overhead that is added to each protected packet.

7. Conclusions

This paper has presented an assessment of the communication overhead of IPsec and has evaluated the feasibility of deploying IPsec on handheld mobile devices and the UMTS radio interface protocol architecture, which are characterized by limited processing and bandwidth resources, respectively. In order to achieve these goals, we have first quantified the processing overhead of IPsec in terms of the number of CPU cycles required for performing security assuming the most prominent security algorithms (i.e. DES, 3DES, AES, HMAC-MD5 and HMAC-SHA-1). Apart from the processing overhead, the application of IPsec adds an additional space overhead by increasing the size of the final transmitted packets. The increase owes to the extra security fields encapsulated with the user data in the final packets, and has the natural effect of increasing both the packet transmission time and the final output rate. We have quantified the space overhead of IPsec, and we have given formulas for expressing the size of the protected packets as a function of the size of the user data packets.

The analyzed overheads have been incorporated in a simulation model that represents an IPsec-equipped UMTS mobile device. We simulated fifty-three different security scenarios, and studied the performance in terms of throughput, packet delay, and rate increase of the protected data from the application of IPsec. It has been observed that the more "lightweight" security schemes like HMAC-MD5, HMAC-SHA-1, DES, and the AES encryption with a 128-bit key do not degrade the throughput of the system and that they hardly have an impact on the total delay, as long as the MS processing rate of 100 MIPS or greater. On the other hand, more complex security schemes like 3DES and AES with a 192-bit or 256-bit key provide for an increased resistance against attacks, but pose higher processing requirements and, thus, reduce the system throughput, when the MS processing rate is below 300 MIPS. Moreover, these algorithms have stronger impact on the total delay values and under sufficiently high user data rates lead to excessive delay values. Combining confidentiality with authentication services by adding HMAC-MD5 or HMAC-SHA-1 to AES, DES and 3DES increases even more the strain on the processor. This extra strain is, however, relatively small as compared to these imposed by the encryption schemes.

The presented simulation results quantify the differences in communication overhead of the various algorithms and security options of IPsec, in the considered resource constrained environment. These results can be used for assisting IPsec adoption decisions, and in particular, the choice of a specific security scheme. Apart from the simulation model, we developed a simple analytic model of an IPsec-equipped MS. This model, although approximate, captures the qualitative behavior of an actual system, and thus, can be used for quick investigations of possible new IPsec deployment scenarios avoiding the need for laborious simulations.

As part of our on-going work we study IPsec performance issues and their connection to user mobility and key exchange mechanisms.

Acknowledgement

Work supported in part by the project CASCADAS (IST-027807) funded by the FET Program of the European Commission and the Program "PYTHAGORAS II: Support of Universities' research groups" co-funded by the "Operational Programme for Education and Initial Vocational Training" (O.P. "Education") and the European Social Fund.

Appendix A

The following table presents the source data and the confidence intervals for Fig. 6. Delay values are derived for different data rates under an assumed processing rate of 100 MIPS and different security configurations (DES, DES + SHA-1, AES(128)Enc and AES(128)Enc + SHA-1).

Delay values

Rate	No security			DES			DES + SHA-1			AES(128)Enc			AES(128)Enc + SHA-1		
	Low	Mean	High	Low	Mean	High	Low	Mean	High	Low	Mean	High	Low	Mean	High
144	57	59	61	75	78	81	77	80	83	81	83	85	82	84	86
384	160	164	168	215	220	225	219	224	229	224	228	232	228	232	236
512	232	240	248	311	318	325	317	325	333	324	331	338	334	340	346
640	334	347	360	433	444	455	443	456	469	457	467	477	471	482	493
768	504	519	534	638	653	668	653	669	685	670	684	698	690	706	722
860	757	775	793	1022	1043	1064	1049	1071	1093	1076	1093	1110	1132	1152	1172
900	938	957	976	1229	1254	1279	1259	1286	1313	1292	1312	1332	1345	1380	1415

References

[1] K.H. Cheung, J. Misic, On virtual private networks security design issues, Computer Networks 38 (2) (2002) 165–179.

[2] O. Elkeelany et al., Performance analysis of IPsec protocol: encryption and authentication, In: IEEE Communications Conference (ICC 2002), 2002, pp. 1164–1168.

[3] S. Miltchev, S. Ioannidis, A. Keromytis, A study of the relative costs of network security protocols, in: Proceedings of USENIX 2002 Annual Technical Conference, Monterey, CA, June 2002.

[4] P. Ganesan et al., Analysing and modelling encryption overhead for sensor network nodes, in: Proceedings of WSAN'03, San Diego, California, USA, September 2003.

[5] S. Kent, R. Atkinson, Security Architecture for the Internet Protocol, RFC 2401, November 1998.

[6] S. Kent, R. Atkinson, IP Authentication Header, RFC 2402, November 1998.

[7] S. Kent, R. Atkinson, IP Encapsulating Security Payload (ESP), RFC 2406, November 1998.

[8] W. Stallings, Network Security Essentials: Applications and Standards, Prentice-Hall, 2000.

[9] E. Danielyan, Goodbye DES, Welcome AES, Cisco The Internet Protocol Journal 4 (2) (2001) 15–21.

[10] S. Frankel, R. Glenn, S. Kelly, The AES-CBC Cipher Algorithm and Its Use with IPsec, RFC 3602, September 2003.

[11] R. Phan, Impossible differential cryptanalysis of 7-round Advanced Encryption Standard (AES), Information Processing Letters 91 (1) (2004) 33–38.

[12] D. Bertsekas, R. Gallager, Data Networks, Prentice-Hall, 1992.

[13] US National Bureau of Standards, Data Encryption Standard, Federal Information Processing Standard (FIPS) publication 46-2, December 1993. Available from: <http://www.itl.nist.gov/fipspubs/fip46-2.htm>.

[14] R. Rivest, The MD5 Message-Digest Algorithm, RFC1321, April 1992.

[15] D. Eastlake, P. Jones, US Secure Hash Algorithm 1 (SHA1), RFC 3174, September 2001.

[16] C. Madson, R. Glenn, The Use of HMAC-MD5-96 within ESP and AH, RFC 2403, November 1998.

[17] C. Madson, R. Glenn, The Use of HMAC-SHA-1-96 within ESP and AH, RFC 2404, November 1998.

[18] National Institute of Standards and Technology (NIST), Advanced Encryption Standard (AES), Federal Information Processing Standard (FIPS) publication 197, November 2001. Available from: <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>.

[19] F. Granelli, G. Boato, A novel methodology for analysis of the computational complexity of block ciphers: Rijndael, Camellia and Shacal-2 compared, in: 3rd Conference on Security and Network Architectures (SAR'04), June 2004. Available from: <http://eprints.biblio.unitn.it/archive/00000514/01/DIT-04-004.pdf>.

[20] J. Daemen, V. Rijmen, The Design of Rijndael, Springer, 2002.

[21] C. Lu, S. Tseng, Integrated Design of AES (Advanced Encryption Standard) Encrypter and Decrypter, in: Proceedings of the IEEE International Conference on Application-Specific Systems, Architectures, and Processors (ASAP'02), 2002.

[22] ETSI, Universal Mobile Telecommunication System (UMTS); Selection Procedures for the Choice of Radio Transmission Technologies of the UMTS, Technical Report TR 101 112 v3.2.0,1998.

[23] J. Korhonen, Introduction to 3G Mobile Communications, Artech House, 2003.

[24] 3GPP TS 25.321 (v3.15.0), Medium Access Control (MAC) protocol specification, release '99, March 2003.

[25] 3GPP TS 25.322 (v3.6.0), Radio Link Control (RLC) protocol specification, release '99, March 2001.

[26] 3GPP TS 25.323 (v3.4.0), Packet Data Convergence Protocol (PDCP) specification, release '99, March 2001.

[27] ARM microprocessor solutions from ARM Ltd. Available from: <http://www.arm.com/products/CPUs>.

Christos Xenakis (xenakis@di.uoa.gr) received his B.Sc. degree in computer science in 1993 and his M.Sc. degree in telecommunication and computer networks in 1996, both from the Department of Informatics and Telecommunications, University of Athens, Greece. In 2004 he received his Ph.D. from the University of Athens (Department of Informatics and Telecommunications). From 1998 to 2001 he was with a Greek telecoms system development firm, where he was involved in the design and development of advanced telecommunications subsystems for ISDN, ATM, GSM, and GPRS. Since 1996 he has been a member of the Communication Networks Laboratory of the University of Athens. Currently, he is a visiting Lecturer in the Department of Informatics and Telecommunications, University of Athens. He has participated in numerous projects realized in the context of EU Programs (ACTS, ESPRIT, IST). His research interests are in the field of mobile/wireless networks and security. He is the author of over 20 papers in the above areas.

Nikos Laoutaris received the Ph.D. degree from the Department of Informatics and Telecommunications of the University of Athens, Greece, in 2004, for his work in the area of Content Networking. He also holds an M.Sc. degree in Telecommunications and Computer Networks (2001) and a B.Sc. degree in Computer Science (1998), both from the same department. His main research interests are in the analysis of algorithms and the performance evaluation of Internet content distribution systems (CDN, P2P, web caching) and multimedia streaming applications. He is currently a Marie Curie Outgoing International post-doctoral fellow at the Computer Science Department of Boston University.

Lazaros Merakos (merakos@di.uoa.gr) received the Diploma in electrical and mechanical engineering from the National Technical University of Athens, Greece, in 1978, and the M.S. and Ph.D. degrees in electrical engineering from the State University of New York, Buffalo, in 1981 and 1984, respectively. From 1983 to 1986, he was on the faculty of Electrical Engineering and Computer Science at the University of Connecticut, Storrs. From 1986 to 1994 he was on the faculty of the Electrical and Computer Engineering Department at Northeastern University, Boston, MA. During the period 1993–1994 he served as Director of the Communications and Digital Processing Research Center at Northeastern University. During the summers of 1990 and 1991, he was a Visiting Scientist at the IBM T.J. Watson Research Center, Yorktown Heights, NY. In 1994, he joined the faculty of the University of Athens, Athens, Greece, where he is presently a Professor in the Department of Informatics and Telecommunications, and Director of the Communication Networks Laboratory (UoA-CNL) and the Networks Operations and Management Center. His research interests are in the design and performance analysis of broadband networks, and wireless/mobile communication systems and services. He has authored more than 150 papers in the above areas. Since 1995, he is leading the research activities of UoA-CNL in the area of mobile communications, in the framework of the Advanced Communication Technologies & Services (ACTS) and Information Society Technologies (IST) programmes funded by the European Union (projects RAINBOW, Magic WAND, WINE, MOBIVAS, POLOS, ANWIRE). He is chairman of the board of the Greek Universities Network, the Greek Schools Network, and member of the board of the Greek Research Network. In 1994, he received the Guanella Award for the Best Paper presented at the International Zurich Seminar on Mobile Communications.

Ioannis Stavrakakis: Diploma in Electrical Engineering, Aristotelian University of Thessaloniki, (Greece), 1983; Ph.D. in EE, University of Virginia (USA), 1988; assist. Prof. in CSEE, University of Vermont (USA), 1988–1994; assoc. prof. of ECE, Northeastern University, Boston (USA), 1994–1999; assoc. prof. of Informatics and Telecommunications, University of Athens (Greece), 1999–2002 and prof. since 2002. Teaching and research interests are focused on resource allocation protocols and traffic management for communication networks, with recent emphasis on peer to peer, wireless, sensor and ad hoc networking. His past research has been published in over 130 scientific journals and conference proceedings and was funded by NSF, DARPA, GTE, BBN and Motorola (USA) as well as Greek and European Union (IST) Funding agencies. He has served repeatedly in NSF and IST research proposal review panels and involved in the organization of numerous conferences sponsored by IEEE, ACM, ITC and IFIP societies. He is a senior member of IEEE, a member of (and has served as an elected officer for) the IEEE Technical Committee on Computer Communications (TCCC) and the chairman of IFIP WG6.3. He has served as a co-organizer of the 1996 International Teletraffic Congress (ITC) Mini-Seminar, the organizer of the 1999 IFIP WG6.3 workshop, a technical program co-chair for the IFIP Networking'2000, EWC'04 and IFIP WiOpt'05 conferences, the Vice-General Chair for Networking'2002 conference, the organizer of the COST-IST(EU)/NSF(USA)-sponsored NeXtworking'03 and the Workshop on Autonomic Communications (WAC2005). He is an associate editor for the IEEE/ACM transactions on Networking, the ACM/Baltzer Wireless Networks Journal and the Computer Networks Journals.