Security versus Performance Requirements in Data Communication Systems

Vasilios Zorkadis

FZI Forschungszentrum Informatik an der Universität Karlsruhe, Germany.

Abstract. The research activities in secure computer networks have paid little attention to the tradeoff between security and other quality requirements of the communication service. This paper aims to introduce performance aspects regarding secure computer networks. First, we attempt to quantify the tradeoff between security and performance in secure data communication systems by means of queueing theory. Our second target is to reduce the performance degradation caused by the security mechanisms and protocols. For this purpose, optimization concepts are proposed. The key points in the optimization concepts are: preprocessing, message segmenting and compression. They have to be integrated or considered in secure communication protocols to improve their performance characteristics. Preprocessing aims to exploit the idle periods of the system (e.g., computer or special crypto-chip), to take the stochastic nature of such communication processes into consideration, e.g., using the OFB-mode for generating (pseudo) random bit sequences after connection establishment. Segmenting is proposed for long messages in order to better exploit the pipeline nature of communication systems. Also, compression is discussed as a means to further improve the performance measures of secure communication.

1 Introduction

Computer network systems provide data transfer services for computer systems. Further requirements such as performance, security, and reliability characterize the quality of the transfer service. However, these requirements affect each other such that a decision has to be made for cases in which all or some of the requirements are desired but cannot be fulfilled. As an example, consider the case when confidentiality over a 155 Mb/s network is desired, but no encipherment devices of such speed are available. Another example could be firewalls (e.g. Interlock, 500 Kbit/s encipherment and 1,2 Mb/s authentication [13]) whose security functions (authentication and encipherment) are the system bottlenecks when communicating over a T1 link. Furthermore, the additional traffic generated through the firewalls must be considered since its performance impact could be prohibitive for their employment. In this section we will introduce briefly the most important security service classes. In the second section we will study the tradeoff between security and performance, and in the third section we will describe and analyze the optimization concepts.

There are five classes of security services in the field of secure data communication systems: authentication, access control, confidentiality, integrity and non-repudiation services. Table 1 shows the classification of the security services according to the OSI Security Architecture 7498-2 [11].

Authentication services refer to one-sided or mutual authentication of peer entities as well as to the authentication of the data origin. They mostly involve trusted third parties and are based on symmetrical or asymmetrical methods. They can be simple like the exchange of unprotected passwords or complex like the three-way-authentication in the Authentication Framework [12]. We refer to [3,9] and the references therein for information about authentication, authentication services and mechanisms, and authentication servers like Kerberos. Authentication of the peer entities takes place before the message exchange begins and data origin authentication during the data transfer phase. Authentication affects the performance of a data communication association this way at its beginning and during the data transfer phase.

Tab 1. Classification of the OSI Security Services

Security services	Subclasses
Authentication	Peer entities Data origin
Access Control	Communication services, Hosts, Networks, Subnets etc.
Confidentiality	Data (connection-oriented,-less, selective field) Traffic-related data
Integrity	Data (connection-oriented,-less, selective field)
Non-repudiation	Data (origin and/or delivery)

Access control services may be employed in the connection establishment phase with connection-oriented communication and for each message with connectionless communication. These services are local in the form of outgoing access control and remote with respect to different security domains and intermediate systems through which the communication is routed. Access Control is based on authentication and integrity, in addition to authorization table look-ups (e. g., checking the related information in the access control lists). From our performance point of view, it is apparent that access control services introduce computing costs either at the connection establishment phase or during the data transfer phase.

Confidentiality services may be employed for whole or parts of messages as well as for traffic-related information, as who communicates with whom and when and how many messages are exchanged and how long they are (traffic analysis). We refer to [1,7] for further information on efficient concepts for anonymous networks. Confidentiality services are based on cryptographic algorithms. From our performance point of view, the confidentiality services always impose processing costs at the end systems (and maybe at the intermediate nodes when link-by-link encipherment is employed). Further costs are related to the key exchange and other cryptographic parameters, like initialization variable, cryptographic algorithm names, mode of operation, etc. Also, costs result from expanding the messages being enciphered due to block-oriented modes of operation. The latter costs could be avoided [2].

Integrity services, like confidentiality services, may be employed for whole or parts of messages and message streams. They include the calculation of check sums and the appending of these and message sequence numbers to the messages. Timestamps and challenge/response mechanisms may be used against the replay attack.

The performance costs result from the check sum calculations and the expansion of the message length.

Non-repudiation services provide for one-sided (origin or delivery) or mutual (both origin and delivery) protection of the communicating parties with respect to each other. They always include authentication and integrity services and may include confidentiality as well. They have juridical character and therefore may be of rare usage in non-commercial networks. Their performance costs as the costs for authentication and integrity, and confidentiality services arise from the additional traffic, the processing and the message length expansion due to the several check sums.

2 Security versus Performance

In the introduction, we addressed the performance costs that arise due to the different security mechanisms. Table 2 contains security services and their corresponding performance costs. The performance costs can be local and/or remote according to the security model used, including the specific security mechanisms, such as access control and encipherment concepts, the use of trusted third parties (e.g., Kerberos), the use of firewalls, security-related routing control, etc.

Table 2. Security services and their associated performance costs

Security services	Performance costs
Authentication	Additional traffic arising at the connection establishment phase processing costs during the data transfer phase
Access control	Processing costs at connection establishment or data transfer phase
Confidentiality	Processing costs in the end systems (end-to-end encipherment) or the end and the intermediate systems (link-by-link encipherment)
Integrity	Message length expansion and calculation costs
Non-repudiation	Like authentication and integrity costs with respect to the kind (and confidentiality costs in case of confidential communication)

In connection-oriented communication, the connection establishment and the data transfer phase are of importance, as opposed to the connection release phase, unless a new communication association is immediately requested. The most important performance measures are throughput and delay. In the connection establishment phase, only the delay is reasonable as a performance measure. Both performance measures are of interest in the data transfer phase for the connection-oriented as well as for the connectionless communication.

The performance analysis of the connection establishment phase involves the analysis of the delay that elapses from the time the connection is requested until the time data transfer can begin, and depends on the concrete authentication mechanisms that are employed. Yet the delay for the communication with the authentication server (e.g., Kerberos) has to be considered when trusted third parties are involved in the authentication protocols. The delay depends on the number of authentication messages to be exchanged (one-way, two-way or three-way authentication, number of intermediate systems to be involved, etc.), their length and the computational costs of the security functions to be employed. (encipherment, generation of random bit sequences, like challenge, etc.)

In the following, we will consider only the data transfer phase for performance analyses. As an example of a secure communication protocol, we will briefly describe the S-SNMP (Secure Simple Network Management Protocol [8]). Network management comprises protocols related to fault, accounting, configuration and name, performance, and security management. The security services that are provided by S-SNMP are: data integrity (against duplication, insertion, modification, resequencing, or replays), data-origin authentication (for corroboration of a message's source), access control and data confidentiality. The message formats are as shown in Fig. 1 [8]. The MD5 message-digest algorithm is used to calculate the 128-bit digest and DES for encryption. A D(estination)-timestamp as well as a S(ource)-timestamp are used to simplify clock synchronization and the context parameter to enable access control. PDU is the acronym for Protocol Data Unit and privDest is the destination party identifier. When privacy is required the privDest field remains in plaintext form to enable the receiving entity to determine the privacy characteristics (e.g., which key to use for decryption) of the D-Party. The timestamps are 32 bits long (but the associated fields are 2 octets longer due to encoding-specific reasons) and the other fields may be of variable length (minimum 3 octets).

privDest	digest	D-timestamp	S-timestamp	D-Party	S-Party	Context	PDU

(a) Authenticated but not private

privDest	field of 0 length	D-Party	S-Party	Context	PDU

(b) Private but not authenticated

privDest	digest	D-timestamp	S-timestamp	D-Party	S-Party	Context	PDU

(c) Private and authenticated

Fig. 1. Message formats for S(ecure)-SNMP (version 2)

2.1 A Simple Performance Model of Communication Systems

In the following, we want to model a communication system, first without and then with security mechanisms employed. The model comprises three components: two communicating open systems (OS1 and OS2) and the communication subsystem (CS, see Fig. 2). Each of the ovals OS1, OS2 and CS in the Fig 2 describes a queueing system. As an example consider the communication of two firewalls over a T1 link (ca. 1,5Mbps in USA and 2 Mbps in Europe). The product Interlock [13] provides for 500 Kbps encryption and for 1,2 Mbps data integrity and data origin authentication. We assume that the external arrival process is Poisson distributed with parameter λ

(arrival rate) and the service times are constant with corresponding service rates μ_1 μ_2 and μ_3 respectively. When considering secure communication, the arrival parameter λ remains the same but the service rates, which are now μ_1', μ_2' and μ_3' respectively, change due to the additional processing for encryption and integrity in the gateways, and due to longer messages or packets for the data integrity and data origin authentication in the communcation subsystem.

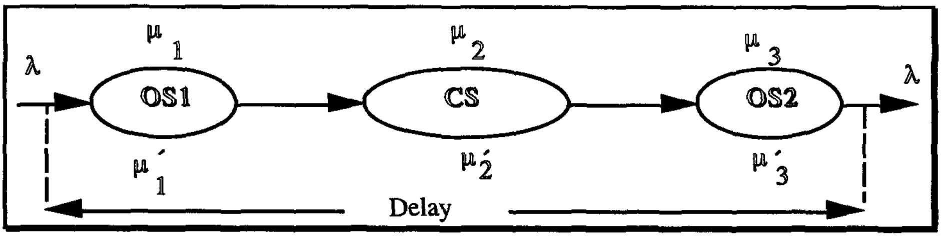

Fig. 2. A simple model of a communication system

2.2 Performance Analysis

The maximum throughput is determined by the bottleneck queue, with parameter $\mu_b = \min_{i\in\{1,2,3\}}(\mu_i)$. The maximum throughput is then equal to μ_b. In the case of our example above, when considering insecure communication, the bottleneck component is the T1 link, and when considering secure communication, the bottleneck component is the gateway (or firewall or open system) due to the encryption or the integrity and authentication functions required.

The component utilizations are $\rho_i = \lambda/\mu_i$. The definition $s_i = \mu_i/\mu_i'$ leads to $\rho_i' = \lambda/\mu_i' = s_i(\lambda/\mu_i) = s_i\rho_i$.

The average delay spent in the system by a packet is equal to the sum of the times spent in each of the queueing systems. For our model (i.e., a tandem net with constant service times and external arrivals Poisson distributed) we can calculate the delay by only considering the bottleneck component as an M/D/1 [6] queueing system regardless of its position in the model chain (i.e., regardless of the actual arrival process of the bottleneck component) and adding to this result just the service times spent in the other components.

Thus, we have the average time T for insecure and T' for secure communication spent in the system as the sum of the time spent in the bottleneck components plus the service times in the other components:

$$\boxed{\begin{aligned} T &= \frac{1}{\mu_b} + \frac{1}{\mu_b}\frac{\rho_b}{2(1-\rho_b)} + \sum_{\substack{i=1\\ i\neq b}}^{3}\frac{1}{\mu_i} = \frac{1}{\mu_b}\frac{\rho_b}{2(1-\rho_b)} + \sum_{i=1}^{3}\frac{1}{\mu_i} \\ T' &= \frac{1}{\mu_b'} + \frac{1}{\mu_b'}\frac{\rho_b'}{2(1-\rho_b')} + \sum_{\substack{i=1\\ i\neq b}}^{3}\frac{1}{\mu_i'} = \frac{1}{\mu_b'}\frac{\rho_b'}{2(1-\rho_b')} + \sum_{i=1}^{3}\frac{s_i}{\mu_i} \end{aligned}}$$

Example. The factors s_i depend not only on the security functions employed, but also on the message lengths. When communicating 60-bytes-long messages, the additional 24 bytes (16 bytes digest plus 8 bytes timestamps) for data authentication result in a 40% expansion of the message length. In this case we obtain: $s_2 = 1{,}4$. For longer messages we obtain smaller values for s_2. According to literature reports on traffic statistics (e.g., [5], these statistics follow the same general pattern of statistics collected on general-purpose networks) 52,8% of the total number of packets have a length of 60-98 bytes, approximately 69% of total packets have a length of 60-138 bytes and ca. 87% of them a length of 60-566 bytes. Although many of the packets that are less than 200 bytes are either acknowledgements or control packets [5], it is obvious that a significant percentage of the traffic consists of small packets. Note that acknowledgements are much fewer than packets sent since connectionless communication does not require acknowledgement and that, with connection-oriented communication, one acknowledgement packet acknowledges more than one user packet. Furthermore, we want to point out that s_2 also depends on the specific communication protocols regarding packet formats. For example, frames in CSMA/CD-based LANS must have a minimum length of 60 bytes along with the protocol-related data as opposed to ATM-based networks, where the communicating data unit is a cell with a constant length of 53 bytes (48 bytes payload). So messages which are less than 60 bytes along with their associated protocol overhead must be padded up to 60 bytes for communication over an ethernet-based network. These padding-bytes can be substituted by the integrity check sum without resulting in communication costs. Therefore, the factor s_2 depends on the communication protocols, the traffic characteristics and the security mechanisms employed.

The message expansion, due to the security mechanisms, leads to longer service times for communication protocol functions, like CRC (Cyclic Redundancy Check), but does not for others, like routing. In order to calculate the new service rates and their associated factors, s_1 and s_3, we must add the computing times for the specific security functions to the new service times for the communication protocol functions.

The following diagram (see Fig. 3) contains the response times for:

1) $\mu_1 = \mu_2 = \mu_3 = 1920$ packets/s, insecure communication for 100 byte long packets,

2) $\mu_1' = \mu_3' = 1920$ packets/s and $\mu_2' = 1548$ packets/s, only data integrity and origin authentication,

3) $\mu_1' = \mu_3' = 625$ packets/s and $\mu_2' = 1920$ packets/s, only encryption for 100 byte long packets,

4) $\mu_1' = \mu_3' = 504$ packets/s and $\mu_2' = 1548$ packets/s, authentication and encryption for 100 (+24) byte long packets,

where in the cases 2 and 3 only the encryption service times of the stations OS1 and OS2 (500 Kbps) are considered.

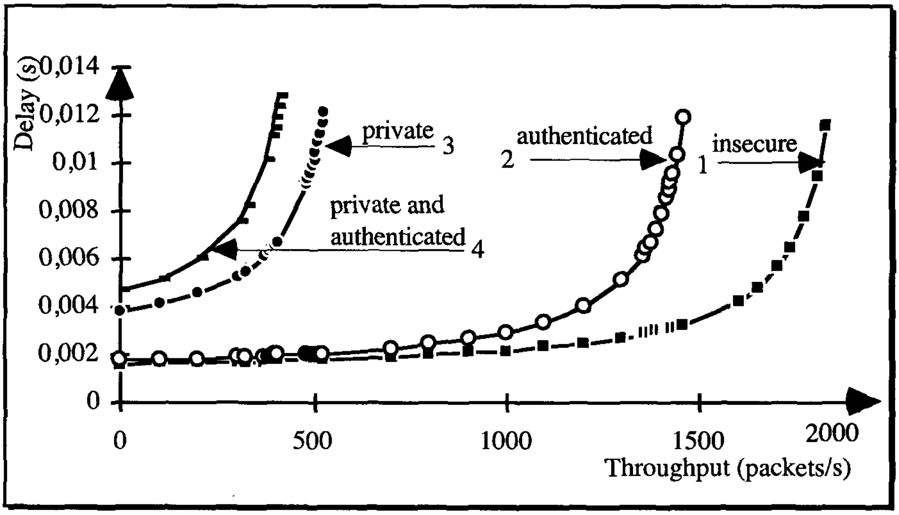

Fig. 3. The average system time as a function of the throughput for insecure (case 1), for authenticated (case 2), for confidential (case 3) and for confidential and authenticated communication (case 4).

For an intuitive explanation of the exponential growth of the response times for $\rho_b \rightarrow 1$ or $\rho_b' \rightarrow 1$, for each case consider that the random arrival process results in bursts of traffic and we must pay an extreme penalty when attempting to run the system near $\rho = 1$ [6].

3 Optimizations

In this chapter we deal with optimization concepts that do not involve substituting the hardware or software components by faster ones. The question in this chapter is: how can we improve the performance measures of secure communication? First, we observe that we cannot increase the maximum throughput, which requires the abovementioned substitution of slow components with faster ones. However, we can improve the delay for equal values of achieveable throughput. Observing, in a differentiated way, the optimization possibilities for integrity, authentication, and encryption, we come to following: For integrity and authentication the goal can be to decline the length of the check sums and other parameters used rather than to reduce the associated processing costs and for encryption to reduce the processing needed. In the following, we deal with optimization concepts for security services requiring encryption.

3.1 Preprocessing

We remind first that the OFB (output feedback) operation mode was mainly intended for applications in which the error-extension properties of CBC (cipher block chaining) and CFB (cipher feedback) were troublesome [2]. A further advantage has been seen in the applicability of bulk encryption to multiple user´s transmissions [5]. As disadvantages were seen an increased sensitivity to bit slippage and a requirement for more complex synchronization procedures [5]. When employing this OFB operation mode, the cryptographic functions are actually used as a pseudo-random bit-string generator. When using OFB or any other strong random number generator for encryption purposes in the application-oriented layers, the above-mentioned disadvantages are no longer present since the transport-oriented layers provide for a reliable error-free message or packet end-to-end transfer. We should also point out that traditional transmission media possess an error bit rate between 10^{-4} and 10^{-6} as opposed to optical transmission media with an error bit rate of ca.10^{-13}. Now, consider the case of when a connection-oriented communication begins. The arrival process of the messages to be processed and sent is random. Idle and busy time intervals alternate. It is the idle times, e.g., user think times, we intend to exploit to preprocess the pseudo-random bit strings when employing OFB-like encryption functions so that we can improve the average delay. In the following, we treat the preprocessing concept mathematically to calculate the expected performance benefit.

We assume, as above, Poisson distributed arrivals with parameter λ and deterministic service times d. We analyze the following case: When no message is in the queue and the last message in service leaves the server (i.e., the queueing system would enter in an idle period), the server begins to preprocess the random bit sequence for the next message to arrive. For simplifying the analysis, we will consider only the case where the server preprocesses the relevant pseudo-random bit sequence for the next message when the system would be otherwise idle. When the random bit sequence for a message is processed before its arrival, the system enters in an idle period. We neglect the time for the XOR-operation.

Analysis. We observe that when a new message arrives, the system is in one of the following states:

1) With probability, say now p, the server is busy with messages that arrived prior to the new message, which has to wait until all of the messages which arrived earlier are processed. In that case the server needs the time d (the same as without preprocessing) to completely serve the new arrived message.
2) With probability $(1-p)$, the server preprocessed either all or part of the processing required for the new arrived message. Thus, its new service time depends on the elapsed time since the last message left the system. It ranges from 0 to d, i.e., 0 when the elapsed time since the last message left the system was longer than d and a value less than d when the elapsed time was shorter than d.

The idle periods are exponentially distributed since we assumed a Poisson arrival process and implicitly an exponential distribution for the interarrival times as they are in an M/G/1 system [6]. Let $g(y)$ be the pdf (probability density function) describing the new service time and its Laplace transform be denoted by $G^*(s)$. The $g(y)$ is given by the following equation:

$g(y) = pu(y-d) + (1-p)f(y)$,

where $u(y-d)$ is the unit impulse function, and

$$f(y) = \begin{cases} \lambda\, \mathrm{e}^{-\lambda(\mathrm{d}-\mathrm{y})} + e^{-\lambda d} u(y), & 0 \le y \le d \\ 0, & otherwise \end{cases}$$

Let us now deal with the probability p. From the state description above, it is obvious that p is the time percentage the system serves a message already present (without the preprocessed part), which equals λ multiplied by the new average service time d'. Thus, we can first calculate p and then d'.

$$p = d'\lambda = \lambda\left[pd + (1-p)\int_{0^-}^{d} yf(y)dy\right],\ d' = \begin{cases} \dfrac{d - \frac{1}{\lambda}\left(1 - e^{-\lambda d}\right)}{e^{-\lambda d}} & ,0 < \lambda \le \dfrac{1}{d} \\ 0 & ,\lambda = 0 \end{cases}$$

We obtain the second moment of the new service time $\overline{Y^2}$ by applying one of the following formulas:

$$\overline{Y^2} = \int_0^d y^2 g(y)dy \text{ or } \left.\frac{d^{(2)}\, G^*(s)}{d\, s^2}\right|_{s=0} = (-1)^2\, \overline{Y^2},$$

$$\overline{Y^2} = \lambda\, \mathrm{d'd}^2 + (1 - \lambda\, \mathrm{d'})\left(d^2 - \frac{2}{\lambda^2}\left(d\lambda - 1 + e^{-\lambda\, \mathrm{d}}\right)\right)$$

In the calculation of the waiting time, we must pay attention to the fact that when a new message arrives at the system, the number of messages waiting in the queue and their service times are independently distributed. The service times are then equal to d for all waiting messages without preprocessing, since there would be no idle period for preprocessing between subsequent messages. Hence, this leads to the following, when taking expectations:

$$\begin{aligned} E\{W_i'\} &= E\{R_i'\} + E\left\{\sum\nolimits_{j=i-N_i}^{i-1} E\{X_j = d \mid N_i' \ne 0\}\right\} \\ &= E\{R_i'\} + dE\{N_i'\} \end{aligned}$$

where we denote [14]

W_i': The waiting time in queue of the i-th message.

R_i': The residual service time seen by the i-th message.

X_j: The service time of the j-th message.

N_i': The number of messages found waiting in queue by the i-th message.

Taking the limit as $i \rightarrow \infty$ we obtain

$$W' = R' + dN_Q' = R' + d\lambda\, W = \frac{R'}{(1 - d\lambda)}$$

The mean residual time can be obtained as usual for M/G/1-systems: $R' = \lambda \overline{Y^2}/2$. Now, we can proceed with our performance analysis as we do when analyzing M/G/1 queueing systems. The delay T without and T′ with preprocessing are given by the following equations. We obtain these by means of the Pollaczek-Khinchin mean-value formula [6] and by means of the formula for the waiting time W', respectively:

$$T = d + \frac{\lambda d^2}{2(1 - d\lambda)}, \quad T' = d' + \frac{\lambda\, \overline{\mathrm{Y}^2}}{2(1 - \mathrm{d}\lambda)}$$

The calculation of the performance benefit in terms of mean values leads to the following result:

$$T - T' = (d - d') + \frac{\lambda\left(d^2 - \overline{Y^2}\right)}{2(1 - \mathrm{d}\lambda)} = d$$

This last result leads to a simplified formula for T' as follows:

$$\boxed{T' = \frac{\lambda d^2}{2(1 - d\lambda)}}$$

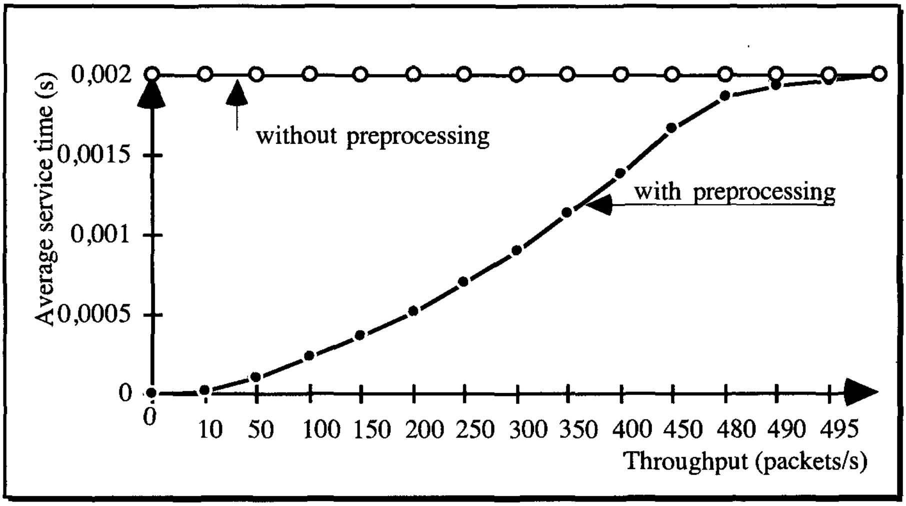

Fig. 4. The average service time with and without preprocessing as a function of the throughput.

Example. The Fig. 4 and 5 show the performance benefit we achieve when applying preprocessing for various utilization values and for $d = 2$ ms. As we expect for $\rho = 0$, the new average response time is zero and for $\rho = 1$ we have no performance benefit. However, for utilizations $\rho \neq 1$ we can achieve a delay reduction with preprocessing (which equals the preprocessable part of the service time d), e.g., for $\rho = 0{,}6$ of ca. 60%, for $\rho = 0{,}86$ of ca. 25% and for $\rho = 0{,}96$ of ca. 8%.

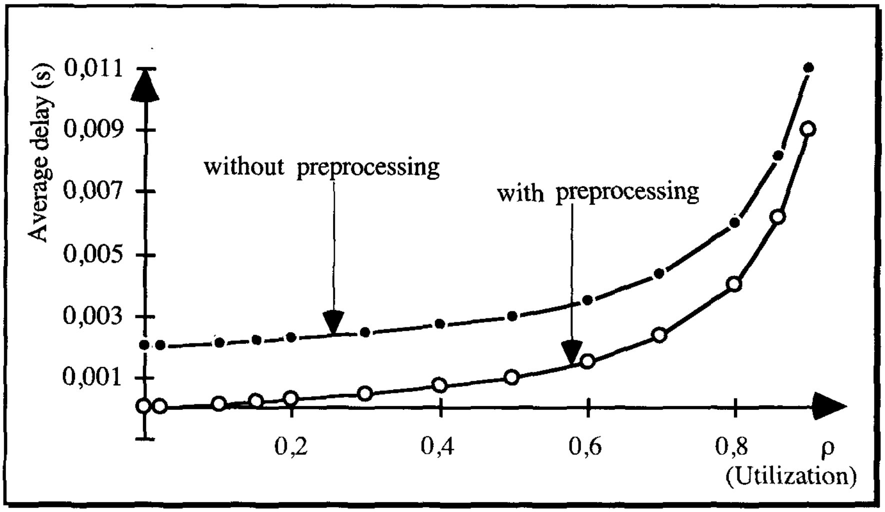

Fig. 5. The average delay with and without preprocessing as a function of the utilization

3.2 Message Segmenting

By communicating long messages and employing encryption in one of the application oriented layers, we can achieve a performance improvement when segmenting the messages in small data units. As an analogon consider packet versus message switching [10]. Associated with the segmenting is additional overhead which must be considered to determine the length of the data units. The performance benefit results from the fact that, while the last part of a message can stil be in service in the bottleneck component (i.e., cryptographic function of the sender), other parts can be taken into service by the subsequent system components (i.e., in the intermediate nodes or in the destination.)

3.3 Message Compression

Also, compression may be used in combination with message segmenting to further improve the performance measures when communicating long messages. The possible benefit is obvious and its analysis straightforward.

4 Conclusion

In this paper we introduced queueing theory in the performance study of secure data networks, and showed and quantified the tradeoff between security and performance. In order to improve the performance of secure communication protocols we discussed and partially analyzed optimization concepts. Preprocessing, message segmenting and compression functions may be employed to significantly improve the delay character istics of secure communication. Compression functionality is foreseen in the presentation layer but preprocessing and message segmenting mechanisms should be considered in secure protocols of application oriented layers.

Acknowledgements

I wish to thank Prof. W. Zorn and H. Schickle for their useful comments and Michelle Specht and Rosi Charitopoulou for text corrections.

References

1. D. Chaum: Security Without Identification: Transaction Systems to Make Big Brother Obsolete, Communications of the ACM, Oct. 1985, V. 28, No. 10, pp. 1030-1044.
2. D. W. Davies, W. L. Price: Security for Computer Networks, John Willey & Sons, Inc., Second Edition, 1989.
3. D. Gollmann, T. Beth, F. Damm: Authentication services in distributed systems, Computers & Security, 12 (1993), pp. 753-764.
4. M. J. Johnson: Using high-performance networks to enable computational aerosciences applications, Proc. of the IFIP WG6.1/WG6.4 Third International Workshop on Protocols for High-Speed Networks, Stockholm, Sweden, 13-15 May, 1992, pp. 137-152.
5. R. R. Jueneman: Analysis of Certain Aspects of Output Feedback Mode, Proc. of CRYPTO 1982, Advances in Cryptology, Plenum Press 1983, pp. 99-127.
6. L. Kleinrock: Queueing Systems, Volume I: Theory, John Willey & Sons, Inc. 1975.
7. A. Pfitzmann, M. Waidner: Networks without User Observability, Computers & Security, 6 (1987), pp. 158-166.
8. W. Stallings: SNMP, SNMPv2 and CMIP: the practical guide to network management standards, Addison-Wesley Publishing Company, Inc., 1993.
9. J. J. Tardo, K. Alagappan: SPX: Global Authentication Using Public Key Certificates, Proc. 1991 IEEE Computer Society Symposium on Research in Security and Privacy, May 20-22, 1991, pp. 232-244.
10. A. S. Tanenbaum: Computer Networks, Prentice-Hall International Editions, Second Edition, 1989.
11. ISO 7498-2: Security Architecture.
12. CCITT 509: Authentication Framework.
13. ANS CO+RE Systems, Inc.: Interlock 2.1 and ANSKeyRing, (18.08.1993).
14. D. Bertsekas, R. Gallager: Data Netwotks, Prentice-Hall International Editions, 1987.