Computer Security and Networking Protocols: Technical Issues in Military Data Communications Networks

RONA B. STILLMAN AND CASPER R. DEFIORE

Abstract—**Two problems which limit information sharing among computers are fear of compromise of sensitive data (through accidental disclosure or theft), and designing networking protocols which support effective interconnection of heterogeneous systems. These problems are presented in the context of defense systems—approaches to their solution are discussed—and issues and open questions which must be addressed in the future are identified.**

I. INTRODUCTION

INCREASINGLY, separate elements of the defense community (e.g., command and control, logistics, weather services, intelligence) are recognizing the desirability of more effective information sharing and exchange. Technology exists in the form of computers and communications to support such interaction. Mitigating against this natural desire to move toward greater data exchange, however, is the fear of compromise of sensitive data (through inadvertent disclosure or theft), and the limited ability to effectively interconnect heterogeneous computer systems. These problems, i.e., providing secure data exchange and developing effective networking protocols, their impact on military data networks, and approaches to their solution are discussed in the sequel.

II. THE MILITARY SECURITY PROBLEM

Military data communications systems are, by their nature, shared systems. Host computers are shared by local and remote users; access lines are multiplexed; terminals are concentrated and interfaced to the network by terminal controllers; packet switches interface host computers to the network and handle data traveling to and from all users; and gateways handle data traveling among different networks. When users, cleared to various security levels, generate, manipulate, send, and receive data at assorted levels of classification over a shared communications system, the potential exists for security violations and dangerous compromise.

The security problem, then, is to provide a data communications system (hosts, transmission lines, switches, access controllers, gateways) which prevents unauthorized access to classified information, assures the integrity (i.e., prevents unauthorized modification) of information processed, and protects the availability of network resources for authorized use (i.e., prevents denial of service). Authorized users must be distinguished from interlopers, and the latter denied system access (i.e., the authentication problem); authorized users must be protected from each other, for example, from renegades who would exploit extant system flaws to gain access to classified data to which they are not entitled, or from systems programmers who would create (and later exploit) system flaws (trapdoors, trojan horses). The security mechanisms must function under a variety of conditions including routine use, high traffic stress, and degraded (i.e., partially failed) operations. To provide a context for further discussion, the Worldwide Military Command and Control System (WWMCCS) Intercomputer Network (WIN) will be used as an example of an operational military data communications network. WIN is an ARPANET-like packet switching network interconnecting 20-25 Honeywell 6000/GCOS hosts located at sites across the continental United States, Hawaii, Korea, and Europe. WIN is used in support of day-to-day operations, Joint Chiefs of Staff worldwide exercises, and real world crises and emergencies in such areas as nuclear planning, deployment planning, force status monitoring, aerospace surveillance, and strategic airlift management.

To date, the security problem has been addressed piecemeal, on a subsystem rather than system-wide basis. In the WIN, equipment is shielded to prevent electromagnetic emanations. Transmission lines are bulk encrypted using manually keyed crypto devices at both ends. Encrypting the data vitiates wire tapping; encrypting the header (which includes source address, destination address) precludes analysis of the pattern of traffic flow.

Beyond the crypto devices, in the hosts, packet switches, and network front ends, information is processed in the clear. Over the past decade, many attempts (primarily focused on host machines) have been made to implement software separation and access control mechanisms. At first, these mechanisms were retrofitted into existing operating systems. These "reinforced" systems were broken with such ease and regularity that it became apparent that further attempts to attain security by augmenting a system would be fruitless. To compensate for the lack of logical controls available to provide security, WIN has substituted physical and procedural controls to reduce the threat. Access to terminal and computer rooms is restricted by guards, cipher locks, etc. No dial-up terminals are serviced. All users, regardless of the actual sensitivity of their work, are cleared to the highest level of data processed by the system (so-called "system high" operation). Note that all this does nothing to prevent a user from obtaining access to data to which he/she is not entitled. It simply increases confidence that he will neither attempt to subvert the system nor misuse classified information obtained as a result of system defects. Where economy or policy dictate that all users cannot

Manuscript received January 15, 1980.

R. B. Stillman is with the Office of the Chief Scientist-Associate Director, Technology, Defense Communications Agency, Washington, DC 20305.

C. R. Defiore is with the Data Communications Engineering Branch, Defense Communications Engineering Center, Reston, VA 22090.

U.S. Government work not protected by U.S. copyright

be cleared to system high, users operating at different security levels are separated in time by "periods processing." The computer system is run at system high, level X, for a portion of the day, then is sanitized (users disconnected, memory cleared, removable media replaced), and initialized anew to run system high, level Y. All of this—the locks, the guards, the clearing of users to levels above their actual needs, the rationing of computer resources—is inconvenient and expensive.

These drawbacks are exacerbated in a networking environment. The WIN, and all participating hosts, are run at TOP SECRET, system high. Sites which had operated at lesser security levels, e.g., SECRET, must obtain TOP SECRET accreditation as a prerequisite to WIN connection. That is, they must conduct TOP SECRET background checks of their personnel and institute procedures and controls appropriate to TOP SECRET operation. During periods of SECRET, CONFIDENTIAL, or UNCLASSIFIED operation, a site is disconnected from the WIN. Clearly, these procedural approaches to secure operation run counter to the networking goal of facilitating user interaction.

III. Some Technical Approaches to Providing Security

Given the history of failure of retrofitted security mechanisms and the complexity of typical operating systems, intelligent observers can no longer be convinced that a system is secure by an argument that consists of running an arbitrary set of test cases successfully followed by a broad appeal to intuition. The problem is twofold.

1) Since even simple programs may have an (effectively) infinite input domain and an extraordinarily large number of execution paths, exhaustive testing is impractical.

2) Since the paramount considerations in designing a system are usually production speed, running efficiency, and/or minimal use of storage, and *not* clarity of logic or simplicity of structure, it is unlikely that the system and its behavior can be understood in any meaningful way.

What is required is a mathematical formalism within which well-designed programs can be rigorously proven to be secure.

A security kernel is an access monitor, a mechanism implemented in hardware and software which mediates every access request and carries out only those which are consistent with security policy. The approach to designing and implementing a kernel and proving it secure [7]-[9] is as follows.

1) "Secure behavior" is formally defined (military nondiscretionary security policy is defined in [2]).

2) A mechanism implementing security policy—the kernel—is formally specified in a nonprocedural language. The specification resembles a finite state machine, describing the kernel as a set of states (one of which is the initial state, and one or more of which are final states), a set of state variables, and a set of transforms which moves the machine from state to state and changes the values of the state variables in the process.

3) The formal specification is proven mathematically (using the first-order predicate calculus) to describe a system that behaves securely. The proof is by induction: the kernel is shown to be secure in its initial state, each transform is shown to preserve security, and hence the system is proven secure.

4) The kernel is implemented using a methodology that supports refining the design in layers of increasing detail, with rigorous demonstration (if practical, mathematical proof) that security is preserved from one layer of refinement to the next (Fig. 1). In theory, the process culminates in source code. Since the difficulty of proving anything about a program (or program specification) quickly becomes more difficult with increasing program length [5], [6], [13], the goal is to design a small simple mechanism (ideally, the minimum mechanism necessary) to implement the security policy.

Several state-of-the-art efforts sponsored by the Department of Defense are underway to design and verify secure systems (e.g., Defense Communications Agency's AUTODIN II packet switches and terminal access controllers and DARPA's Kernelized Secure Operating System, a general-purpose time-sharing operating system being built on a PDP-11/70). In addition, the Office of the Assistant Secretary of Defense for Command, Control, Communications, and Intelligence is chairing a DoD consortium to foster the sharing of security technology with the private sector, and thereby encourage the development of secure commercial systems. However, no such systems are yet in operational use. As a result, many important issues are still to be addressed. These include determining the impact of secure system design on the following.

1) System performance, i.e., how much slower does it run, and under what conditions?

2) Development cost, i.e., how much more expensive will it be to build once the technology is widely understood? How expensive is verification and to what level of system description can it be taken? With the cost of hardware falling rapidly and the cost of software not falling at all, can hardware separation be substituted for software protection mechanisms? How, where, and under what circumstances?

3) Maintenance cost, i.e., how often is reverification required in the course of normal system maintenance, and how extensive and expensive is it?

4) Operational flexibility, e.g., can a packet switch be downline loaded in a secure manner? Can a classified connection be preempted, suspended, and restarted securely?

5) System management, e.g., what administrative procedures are required for system certification and recertification as the system evolves?

Rather than attempting to separate multilevel users by monitoring and controlling data accesses, end-to-end encryption attempts to disguise the data at the source, maintain them in unintelligble form all along the communications path, and decrypt them only at the destination.

End-to-end encryption differs from link encryption in some important ways.

1) Under link encryption all bits, header as well as data, are encrypted; under end-to-end encryption, only the data are encrypted.

2) Link encryption is effective only on point-to-point links equipped with crypto units; end-to-end encryption, maintaining clear-text headers, is intended for use over shared

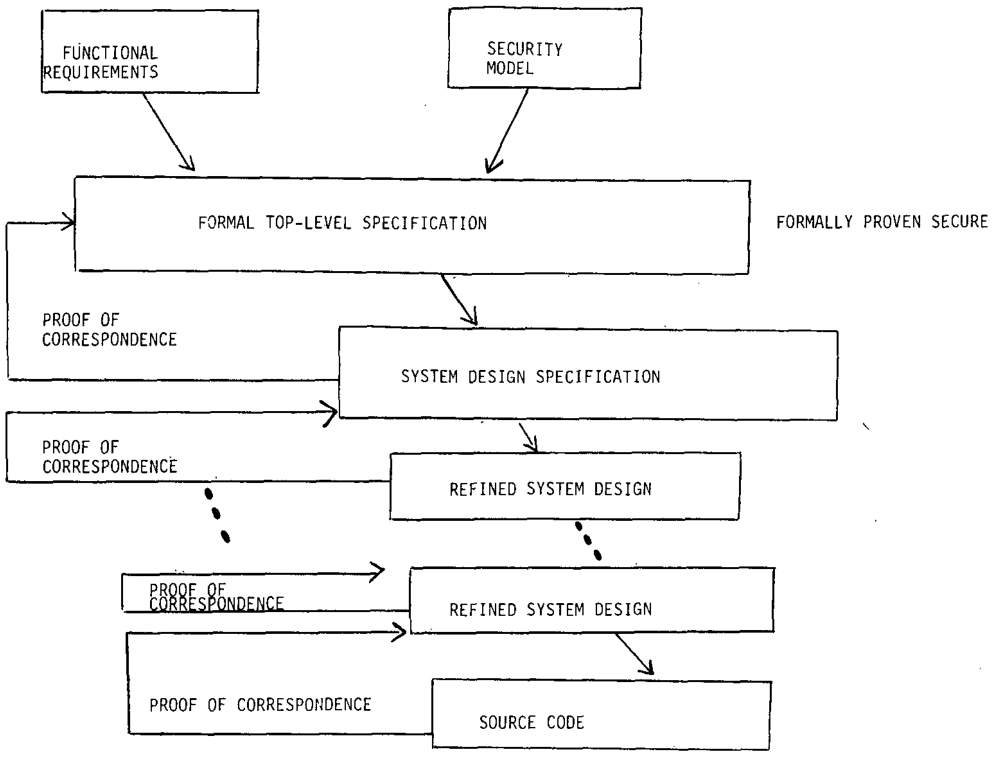

Fig. 1.

channels. (Of course, it is possible to superencrypt by providing both link and end-to-end encryption.)

3) Under link encryption, all processing units (e.g., source, tandem, and destination switches) handle clear-text; hence, all are targets for security attack; under end-to-end encryption, intermediate processing units handle only encrypted data.

4) Under link encryption, all logical channels over the link are encrypted under a single key; under end-to-end encryption, each logical channel may be encrypted under its own key. (This depends, of course, on how the "ends" of the connection are defined. If the ends are sender and receiver processes, then each logical connection will be encrypted under its own key. If the ends are host machines on the network, then all logical connections between a pair of hosts will share an encryption key.)

In end-to-end encryption approaches, security generally rests on the secrecy of the key rather than on the secrecy of the encryption algorithm. Two broad research thrusts in end-to-end encryption are hidden key systems and public key systems. In the former, a single key is used for both encryption and decryption; hence, new keys must be generated and distributed for each pair of users (in practice, for each use). A major challenge in hidden key systems is to design and implement effective and economical key management and distribution mechanisms. In public key systems, there is no known computational relationship between the encryption key and the decryption key, i.e., one cannot be derived logically from the other. As a result, a user of a public key system can make his encryption key public, enabling anyone to send him sensitive information, and be certain that only he can interpret it since the decryption key is held privately. Sender and receiver in a public key system can also authenticate their identities to each other by double encrypting their messages, using first their private decryption key and then their partner's public encryption key [12]. Some open problems in public key systems are to determine how efficient such systems are, and to validate that, for the encryption function used, the encryption and decryption keys are, indeed, computationally independent.

Interesting hybrid systems are being proposed in which a user generates the hidden key to be used for the session (using a random number generator) and transmits the key to his partner using a public key cryptosystem. These systems offer promise of secure and simple key management and distribution within an inexpensive, efficient hidden key encryption system.

Ongoing research efforts are attempting to determine the following.

1) The cost of end-to-end encryption over the system life cycle, and the relationship between encryption costs and level of protection obtained. If a system is encrypted end-to-end, how much additional protection is provided by link encryption? Who needs the added protection and under what circumstances?

2) The effect of end-to-end encryption on communications system performance.

3) The proper place in the systems architecture to perform end-to-end encryption. For example, should it be performed in the host machine, in a network front end device, in a terminal access controller, etc.? Where, in the software architecture, can end-to-end encryption be effectively and economically placed? How does end-to-end encryption impact network and internetwork protocols and functions? Is end-to-end encryption effective for both datagram

and virtual circuit service? (Protocol issues are discussed more fully in the following sections.)

4) If the concept of end-to-end encryption can be extended to include storing and sharing encrypted files. Appropriate key management and distribution methods must be identified and cost assessed.

5) If a broad range of security services can be provided as user options, and if charging algorithms for these services can be developed which are based upon utilization. Examples of security services are link encryption, end-to-end encryption, encryption/decryption of stored files, etc. By allowing a user to select the security services desired for a given session, for example, a normally classified subscriber will be able to communicate with an unclassified subscriber in an unclassified mode.

6) How secure and unclassified networks can best be interconnected.

IV. COMMUNICATIONS NETWORK PROTOCOLS

The rapid growth of computer networking has been supported by recent advances in protocol organization, design, and implementation [11]. A protocol is a collection of rules governing the exchange of data between two network elements. Recent trends in implementing protocols provide for a hierarchial structure known as protocol layering. The basic protocol layers (from lowest to highest) consist of transmission, end-to-end transport, and application-oriented functions (Fig. 2). In such a structure, each level uses the services provided by the next lower level and offers services to the next higher one. No layer makes assumptions about the internal characteristics of any other, and all interaction is via well-defined interfaces. In this way, an implementation at one level can be modified without impacting other levels. Further, different higher levels (e.g., a teleconferencing application and a file transfer application) may call upon the same lower levels (e.g., end-to-end transport) for service.

Hierarchial design coupled with adoption of standards at various levels provide the basis for network survivability, reliability, and interoperability, characteristics which are especially important in military communication networks. Given standard layers of protocol, interconnection between different networks becomes practicable. Ideally, user processes on one network will have access to host services offered on another network, and the physical connection between them may be via other tandem networks.

V. SOME TECHNICAL APPROACHES TO PROTOCOL DEVELOPMENT

The lowest layer of protocol is well established and is defined, for example, in the CCITT Recommendation X.25 [1]. It consists of a physical interface sublevel (X.25 level 1), a transparent frame transfer mechanism sublevel (X.25 level 2), and a packet exchange protocol sublevel (X.25 level 3) which transfers a sequence of packets across the physical interface. One of the problems with X.25 is that there is no guarantee that packets will be delivered reliably across the network. In order to provide end-to-end reliability, a transport protocol is provided. Some data communications systems (e.g., TRANSPAC) combine or extend sublevel 3 to provide end-to-end significance or virtual circuit service. That is, such functions as detecting lost packets, sequencing packets, and connection multiplexing are provided by the network. Other data communications systems (e.g., the DoD common user packet switching network AUTODIN II) do not provide these end-to-end functions as part of the network. Rather, they offer a datagram service in which each packet is treated as a completely independent entity. Datagram service is useful to a growing class of subscribers who want to utilize only the basic data transmission service from a data communications network. These subscribers either do not need an end-to-end protocol or prefer to provide their own. For example, subscribers to a packetized voice service can tolerate occasional packet loss, but even short delays or highly variable ones can cause significant degradation. These subscribers do not need the lost or duplicate packet detection, packet sequencing, or the error correction capability provided by virtual circuit service, and cannot accommodate the delays introduced by such service.

The relative merits of virtual circuit versus datagram service are currently being debated in the networking community. At issue is whether the network should provide the additional services associated with virtual circuits (reordering, retransmission, etc.) to all its subscribers or whether datagram service should be offered as an option. The latter not only permits a subscriber to obtain (and pay for) just the service he requires, but also opens the possibility of interconnecting networks at the datagram service level of protocol, a considerably simpler approach (since no knowledge of the state of the connection is maintained by the network) than interconnecting at the virtual circuit level.

The DoD approach in AUTODIN II is to provide a basic datagram service, and above it in the protocol hierarchy, a virtual circuit service called the Transmission Control Protocol (TCP) [10]. AUTODIN II subscribers can elect to use TCP, can substitute their own connection-oriented protocol, or can use only the datagram service. Currently, DoD is in the process of standardizing TCP as part of a broad program to standardize protocols throughout the Defense community.

Although much is currently being accomplished in the protocol area, many difficult problems remain. Some of the more important issues that relate to military communication systems are the following.

1) While there has been some success in developing international standards for the lowest layer of protocol, similar efforts to standardize higher layers are only beginning. In particular, applications protocols, sometimes referred to as resource sharing protocols [3], have received little attention. Two of the major classes of resource sharing protocols are terminal and file transfer protocols. Standard versions of these protocols are crucial to the effective interconnection of diverse military computer systems, and to the development of additional services such as network mail and distributed databases.

2) The optimal allocation of protocol functions among net-

BASIC LAYER	SUBLEVELS	EXAMPLES *
3. APPLICATIONS FUNCTIONS		FILE TRANSFER PROTOCOL, TELECONFERENCING PROTOCOL, MESSAGE SERVICE PROTOCOL, ETC.
2. END-TO-END TRANSPORT	HOST-TO-HOST PROTOCOL	TRANSMISSION CONTROL PROTOCOL (TCP)
	INTERNETWORK PROTOCOL	INTERNETWORK PROTOCOL (IP)
1. TRANSMISSION	PACKET EXCHANGE PROTOCOL	X.25 LEVEL 3
	FRAME TRANSFER MECHANISM	X.25 LEVEL 2
	PHYSICAL INTERFACE	X.25 LEVEL 1

* These examples are not meant to imply implementation compatability among the designated protocol levels (e.g., X.25 Level 3 and TCP). They are included merely to provide the reader with an understanding of the levels at which certain known protocols belong.

Fig. 2.

work and subscriber elements has not yet been determined. The current thinking is that generally overloaded host computers should perform as few of the network functions as possible, and that such functions should be relegated to a Network Front-End (NFE) processor. The NFE could also provide for direct terminal access to the network. The performance of an NFE-host configuration in an operational environment will be analyzed early in the life of AUTODIN II.

3) The best strategy for attaching an NFE to a host computer has not been determined. In many current systems, the NFE emulates a peripheral device known to the host, e.g., a terminal. While this approach permits network attachment with no substantial modification to the host operating system, generality and efficiency are sacrificed. Where it is possible to make changes to the host operating system software, a much more powerful host interface can be provided [4]. An NFE which interfaces to the host computer using a general and powerful host-to-front end protocol is being standardized for use throughout DoD. The effectiveness of this approach in an operational environment will soon be evaluated.

4) The effects of multiple transmission media on protocols is not yet thoroughly understood. For example, timeouts, flow control, and acknowledgment techniques which are optimal for terrestrial transmission will probably have to be modified for satellite transmission. Considerable research is required before networks can be built which effectively incorporate disparate transmission media. A joint DCA/DARPA experimental program has been initiated in this regard, but results will not be available for several years.

VI. THE INTERPLAY BETWEEN SECURITY AND PROTOCOLS

Network protocols and security are not independent issues. Rather, overall network architecture will be shaped by the interplay between them. For example, if a network is to be multilevel secure by virtue of verifying all "security relevant" software, then any protocol layer which handles data at more than one level of classification is "security relevant" and must be verified. Since software is difficult and expensive to verify and since most protocols are relatively complex, various design and implementation alternatives will be investigated to limit the amount of software to be verified.

Perhaps the protocol can be restructured, separating the portion which handles data at more than one level of classification from the rest; perhaps the protocol can be implemented as a set of single level processes; perhaps the protocol is simply impractical to verify, and is therefore unsuitable for use in a secure network. Similarly, if end-to-end encryption is to be used to provide network security, then any protocol providing a data-related service, i.e., a service which by its nature involves processing data in the clear, must be placed outside the end-to-end encryption boundary. Again, various design and implementation alternatives will be investigated to provide secure service economically.

The nature and impact of the relationship between network security and protocols are not yet fully understood, and clarifying them will be an important area of research for some time to come.

REFERENCES

[1] D. L. A. Barber, "Protocols for intelligent terminals," *Comput. Commun. Rev.*, vol. 9, July 1979.

[2] D. E. Bell and L. J. LaPadula, *Secure Computer Systems*, ESD-TR-73-278, vol. I–III, Electron. Syst. Div., AFSC, Hanscom AFB, MA, Nov. 1973, Nov. 1973, Apr. 1974, AD 770768, AD 771543, AD 780528.

[3] J. D. Day, "Resource sharing protocols," *Computer*, vol. 12, Sept. 1979.

[4] J. D. Day, G. R. Grossman, and R. H. Howe, "WWMCCS host to front end protocols: specifications," Draft, Digital Technol. Inc. (DTI), Document 78012.C-INFE 14, Aug. 1979.

[5] C. A. R. Hoare, "An axiomatic basis for computer programming," *Commun. ACM*, vol. 12, Oct. 1969.

[6] J. King, "A program verifier," Ph.D. dissertation, Dep. Comput. Sci., Carnegie-Mellon Univ., Pittsburgh, PA, 1969.

[7] J. K. Millen, "Security kernel validation in practice," *Commun. ACM*, vol. 19, May 1976.

[8] P. G. Neumann, R. J. Feiertag, K. N. Levitt, and L. Robinson, "Software development and proofs of multi-level security," in *Proc. 2nd Int. Conf. Software Eng.*, 1976.

[9] D. L. Parnas, "A technique for software module specification with examples," *Commun. ACM*, vol. 15, May 1972.

[10] J. Postel, "Transmission control protocol—Version 4," USC/ISI, IEN81, Feb. 1979.

[11] L. Pouzin and H. Zimmermann, "A tutorial on protocols," *Proc. IEEE*, vol. 66, Nov. 1978.

[12] R. L. Rivest, A. Shamir, and L. Adelman, "A method for obtaining digital signatures and public key cryptosystems," *Commun. ACM*, vol. 21, Feb. 1978.

[13] J. A. Robinson, "Review of automatic theorem-proving," in *Proc. Symp. Appl. Math.*, vol. XIX, J.T. Swartz, Ed. Providence, RI: Amer. Math. Soc., 1967.

★

Rona B. Stillman received the B.A. degree in physics from Queens College, New York, NY, the M.A. degree, also in physics, from Yeshiva University, New York, NY, and the Ph.D. degree in systems and information sciences from Syracuse University, Syracuse, NY.

She is currently Assistant for ADP to the Chief Scientist–Associate Director, Technology, Defense Communications Agency, Washington, DC. She has worked in the areas of automated theorem proving, associative processing, software testing and validation, and computer networking.

★

Casper R. Defiore received the B.S. degree in engineering from General Motors Institute, Flint, MI, and the M.S. degree in engineering and the Ph.D. degree in systems and information sciences from Syracuse University, Syracuse, NY.

He is presently with the Defense Communications Engineering Center, Reston, VA, where he is involved in the system engineering of data communications systems. He has worked in the areas of computer architecture, packet switched network protocols, and network security.