A MULTILEVEL SECURE LOCAL AREA NETWORK

Deepinder P. Sidhu
Research & Development
Burroughs Corporation
Paoli, Pennsylvania 19301

Morrie Gasser
The MITRE Corporation
Bedford, Massachusetts 01730

This paper presents a high-level design for a local area network (LAN) that will support subscribers (terminals or hosts) operating at various security levels. Subscribers may be "single-level", which means they are untrusted and can operate at only one security level, or they may be "multilevel" and trusted to operate at a range of security levels [Nibaldi79].

For single-level subscribers, communication is restricted to those at the same security level. This restriction is enforced by trusted interface units (TIUs) used by each subscriber to interface to the LAN, and is based on a security level field in the header of each packet. The TIUs are trusted to enforce and check the security markings in the packets--the hosts or terminals themselves are not.

For multilevel subscribers (a multilevel secure terminal or host), communication is restricted according to the usual security constraints. That is, a multilevel host can transmit at a range of levels between the minimum and maximum. The minimum and maximum are enforced by the TIU for the multilevel host, with the host trusted to choose the specific level of each packet it transmits. Likewise, the multilevel host is trusted to receive packets at the range of its levels and to properly protect the data according to the classification in the packet header. Figure 1 shows a simple multilevel LAN with single-level and multilevel subscribers.

Because the data on the network medium (e.g., coaxial cable) is not encrypted, appropriate physical protection is required. In a broadcast LAN, this would imply that the entire cable and the TIUs would have to be protected to system-high, since all packets on the network are visible at all locations. For subscribers operating at lower security levels (and in less-protected environments) it might not be feasible to protect the medium to system-high at all points, especially since both the TIU-subscriber and TIU-LAN interfaces usually consist of relatively short cables. For example, the physical and procedural controls necessary to provide a DoD Top Secret environment are extremely costly. It would be unreasonable to expect such protection to be required for all the TIUs and the entire LAN medium in a network whose majority of users are unclassified.

To allow for realistic combination of environmental controls we extend the secure LAN architecture to incorporate the concept of separate physical subnetworks whose mediums are each protected to some maximum level that may be less than the maximum level of the entire local network. The subnetworks are connected by "bridges" in such a way that the entire set of subnetworks appear as a single local network to each TIU and subscriber [Clark78]. An example of a LAN composed of several subnetworks is shown in figure 2. Where portions of the medium, TIU-subscriber link, or bridge link must pass through unprotected areas, data is

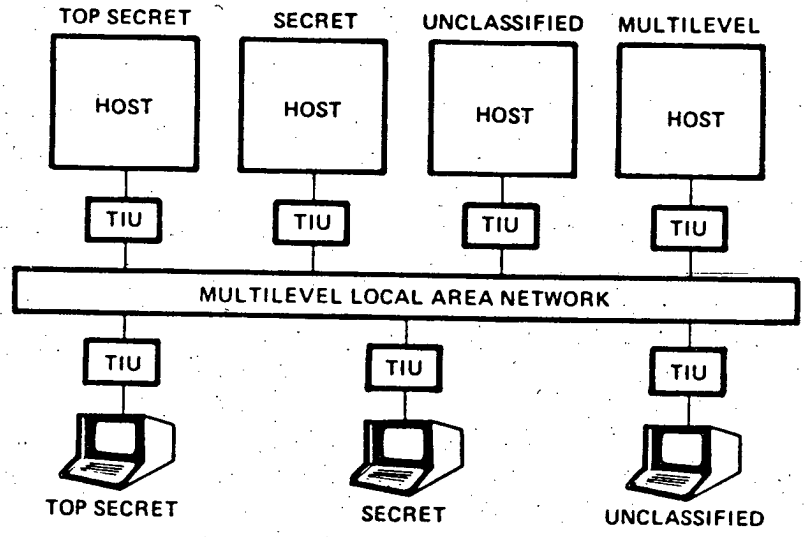

Figure 1. Simple Multilevel LAN

Reprinted from *Proceedings of the 1982 Symposium on Security and Privacy*, 1982, pages 137–143.

EHO208-9/83/0000/0281$01.00 © 1982 IEEE

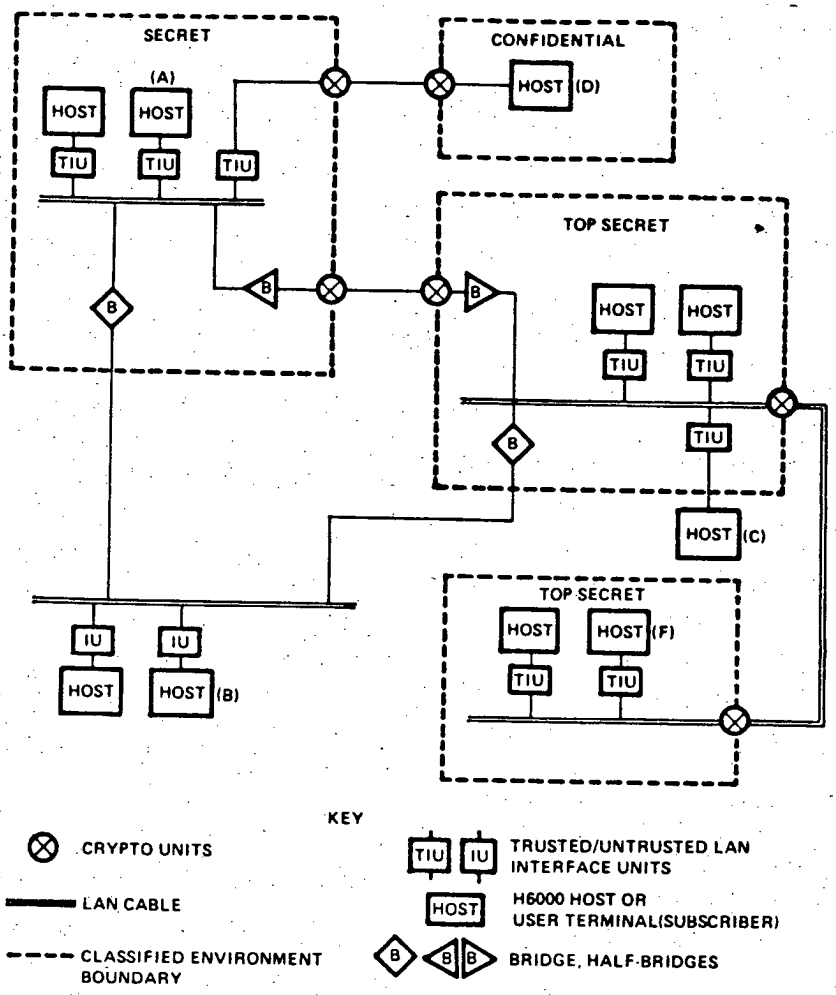

Figure 2. Subnetwork Structure

encrypted using mostly standard link encryption techniques. While this paper does not address encryption as a solution to multilevel data protection, some encryption issues peculiar to this architecture will be discussed below.

The bridges implement a function similar to gateways in wide-area networks but are much simpler. Their job is to route packets between LAN subnetworks with identical protocols. They operate at a level of protocol that makes them transparent to TIUs and hosts or terminals, so that they have no effect on the hardware and software in the TIUs. The bridges perform their routing function based on fixed tables within the bridges and destination addresses in the headers of the packets. They also perform a security check to insure that information from a high level TIU on one subnetwork does not flow to a lower level subnetwork. In this way subnetworks need only be "trusted" (and physically protected) to maintain separation of data within the range of levels of subscribers on that subnetwork. Even unclassified subnetworks can be supported as shown at the bottom of figure 2.

The overall design is intended to be easily implementable with minimal changes to existing off-the-shelf technology and protocols. As such it is felt that it is a practical solution for many installations that have near-term requirements to incorporate a local area network into existing or planned data processing facilities and cannot afford to spend the time or money for more sophisticated long-term options. It is far more flexible than a "system-high" approach where all subscribers must be protected to the highest level [DoD72]. It provides a foundation for multilevel communications at sites that may initially only require communication between single-level entities, but may later upgrade to multilevel hosts and terminals. A proposed implementation would take place in three phases, allowing an initial capability for single-level communication with incremental upgrade to multilevel communication. This phasing matches the anticipated availability of multilevel computers and terminals, where only single-level components are available today, followed by controlled-mode (two-level) hosts, "variable-level" terminals (to be discussed below), and finally multilevel terminals and hosts.

The concept deals with practical matters such as physical protection of the medium, terminals and hosts. It also takes into account the difficulty of certifying large pieces of hardware or software for multilevel operation by reducing to an absolute minimum the components of the system that must be trusted. No "security kernel" or sophisticated trusted mechanisms are required. The design deals with a phased implementation that will satisfy many initial needs in the near future and can later be upgraded, with no disruption of service, for more sophisticated applications as the need for multilevel service increases and a greater volume of traffic must be supported.

The next sections discuss the protocols and the architecture of the TIU and bridge. Some encryption issues are addressed at the end of this paper.

PROTOCOLS

In order to design multilevel security into any network specific protocols must be examined. To insure feasibility of implementation and operation, we are basing our design on an existing protocol that is known to be operational and thus presumably has its bugs worked out. Our goal is to minimize the modifications to the protocol so as not to affect any existing performance studies or implementation techniques. While many of the basic concepts of the approach in this paper are applicable to a number of existing LAN protocols, we have chosen to center our design around the carrier sense multiple access with collision detection (CSMA/CD) protocol that has been proposed for the IEEE standard 802 [IEEE81]. This protocol was chosen because similar protocols are fairly widely (though by no means universally) accepted in industry (Ethernet being the prime example [Ethernet80]). We are not specifying that CSMA/CD is the only protocol that can be used for a multilevel network. However, as design details are presented here it will be apparent where CSMA/CD is specific to this particular design. Certain aspects of the design would have to be modified to use another protocol. When we refer to CSMA/CD in this paper we are specifically referring to the IEEE version, although other versions (e.g., Ethernet) would probably be suitable will little change.

We are not concerned with the issue of whether the LAN medium is a broadband or baseband cable. That is, the physical layer (layer 1 of the ISO reference model [ISO81]) is not an issue for our design, although many aspects of the physical interface (e.g., TEMPEST) must be addressed in an implementation for secure applications. Many of the other aspects of the CSMA/CD protocol not mentioned here remain unchanged from that in the proposed IEEE 802 standard.

Figure 3 shows a simplified format of the IEEE 802 CSMA/CD packet, along with the modified version for our secure LAN. We have subdivided the source and destination address fields into two components, to provide a two-level hierarchical address based on subnetwork number and TIU number. The other

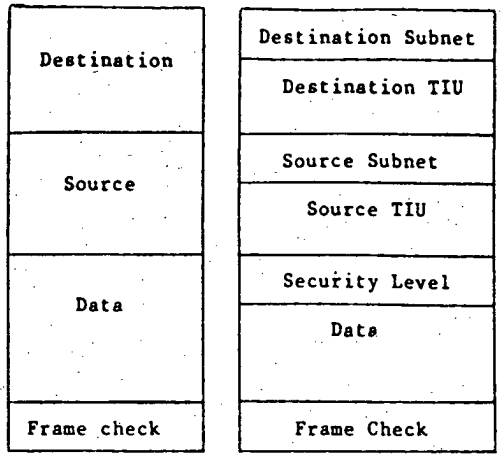

Figure 3. Packet Formats

change is the addition of a security level field at the beginning of the data field. The packet and header length is unchanged, and all the CSMA/CD protocol processing logic is unchanged from that in the standard.

Of course, CSMA/CD is only a low-level link protocol (layer 2 of the ISO reference model), and there are higher layer protocols to be considered in any full implementation. However, our solution is oriented around implementing multilevel security at the link level, with no requirement for any particular protocols at a higher layer. This further minimizes the effect of our approach on any existing software making use of and implementing those higher layers.

Focussing on the link layer alone does have its drawbacks, however. These are seen as minor in an initial scenario where a multilevel LAN would be installed to provide a basic communications capability and also to handle existing security requirements. As the traffic load increases and the type of multilevel processing becomes more sophisticated, the TIUs and bridges on the LAN would be upgraded (in fully compatible manner) so as to provide additional services. These services require the consideration of higher-level protocols, such as the DoD standard Transmission Control Protocol (TCP) with Internet (IP) [Postel81a, Postel81b]. Further discussion of this upgrade capability will be presented in the relevant sections.

TRUSTED INTERFACE UNIT

The TIU is responsible for enforcing the security policy based on the level(s) of its subscriber and the level of packets. TIUs come in three versions, in increasing order of complexity. Initially there would only be the need for single-level TIUs that provide the single-level type of

protection for untrusted subscribers discussed earlier. Another version would provide variable-level operation. This means that the TIU is not permanently fixed to communicate at just one level, but can vary its level based on some human operator action. This type of TIU would allow, for example, a terminal to sometimes operate at one security level (to communicate with a certain set of hosts) and sometimes operate at another security level. Hosts whose levels change due to periods processing would also use a variable-level TIU. Finally, there is a multilevel TIU that properly coordinates with its terminal or host to support full multilevel operation.

Single-level TIU

The trusted interface unit shown in figure 4 allows a single-level subscriber (untrusted host or terminal) to communicate with another subscriber at the same security level, via a local network to which subscribers of several levels are connected. The TIU must be physically protected to the level of network-high, and is designed to reliably isolate the traffic at one particular security level from the traffic at all other levels.

We are using the IEEE 802 standard (CSMA/CD) physical and link layer interface on the network, and envision that off-the-shelf hardware will eventually be available, in the form of a chip or circuit board, that facilitates construction of a microcomputer-based TIU for the CSMA/CD protocol. In our security architecture we have anticipated the functions of such hardware and have made an attempt to use it to simplify the TIU implementation, though our design is by no means dependent on its availability.

The header of the CSMA/CD packet begins with a destination address, followed by the source address. The packet ends with a frame check sequence. The function of the CSMA/CD interface is to recognize valid packets received from the network and to transfer the entire packet into the TIU's memory. (We are describing the interface's function as if it were loading data into a microcomputer memory as a DMA device, although there may be variations on this approach.) When the packet has been successfully received and loaded into memory, the TIU CPU is signalled that a successful DMA transfer has occurred.

The CSMA/CD hardware is assumed to be programmed (or "burned in") with the ability to recognize one particular destination address as its own. As data arrives from the network the first few bytes of header are examined and, if the destination is correct, the remaining data is passed through to TIU memory. If the destination is incorrect, or if a collision is detected, the rest of the packet is ignored (not passed into memory). The CSMA/CD protocol is designed so as to detect all collisions while reading the header of the packet (though this attribute is not currently specified in the standard), so that receipt of a correct destination, coupled with no collision, nearly guarantees that the remainder of the packet in memory is valid (i.e., no collision will occur) and is addressed for the current recipient. Once the packet has been read into memory, it is still possible that a frame-check error will be detected by the hardware as the last byte is read. In this case the TIU CPU is not signalled, so the data, even though now resident in memory, will be ignored.

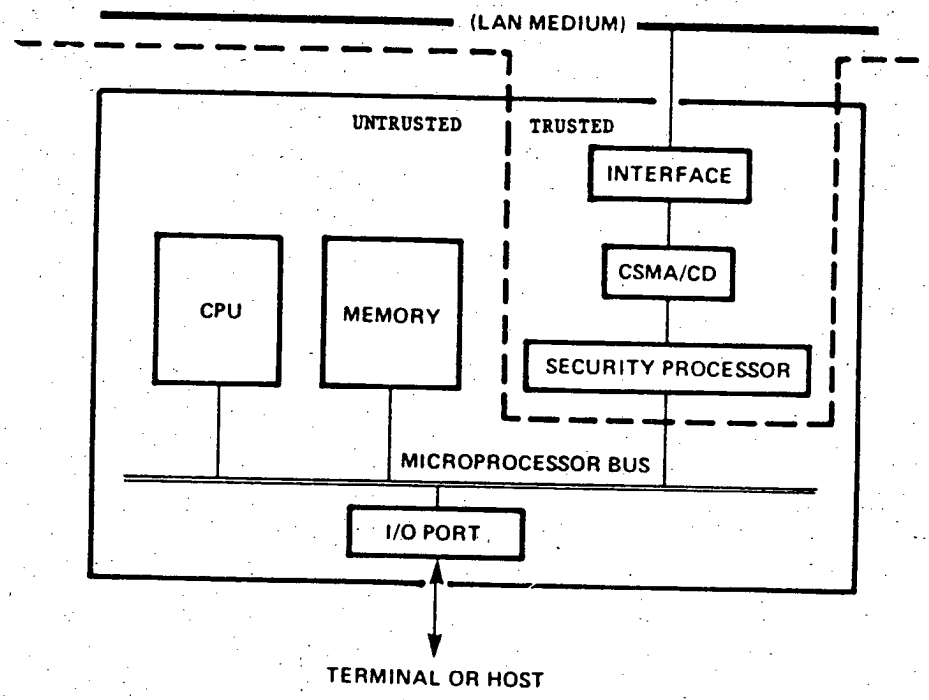

Figure 4. Single-level TIU

On output to the network, a DMA transfer is initiated by the CPU and the interface handles the contention part of the protocol required to get the packet out on the network. It may also, perhaps, handle source address insertion and frame-check computation.

What is important to note is that, between the second field of the header (the source address) and the frame-check sequence, the CSMA/CD interface attaches no particular meaning to the data. For our secure local network we have added an additional field, immediately after the source address, that is the security level of the data. The secure TIU has been designed so as to totally isolate the security-relevant processing (CSMA/CD protocol handling and security field checking) from the remainder of the protocol processing, thus maximizing the flexibility of TIU functions without having to worry about verifying that remainder of the software within the TIU. This concept is particularly important if more complex protocols, such as TCP and IP, are implemented in TIUs. Of course, the security field checking mechanism and many properties of the CSMA/CD protocol handler must be verified.

In the figure we have added a security processor between the CSMA/CD interface and the rest of the TIU, so that there is a distinct trusted/untrusted separation of functions*. On data input from the network, the function of the security processor is to look at the third field of the header, the security level, and only accept the remainder of the packet data if the security level is equal to that of the subscriber. Thus, data will only arrive in the TIU's memory if both the destination and security level are correct. On output to the network, the security processor inserts the subscriber's security level in the packet as the packet is transferred to the network from memory.

We envision the security processor to consist of hardwired logic or perhaps a single-chip computer with an on-board program. This processor has a very simple function since it only processes "good" packets, due to the outboard handling of the contention protocol by the CSMA/CD interface. Also, it need only have the throughput of the subscriber device, not that of the network, since only the subscriber's input and output need be processed. However, depending on the interface characteristics of the CSMA/CD hardware, instantaneous speed might have to be much higher.

Note that, on input from the network, security rules require that no data arrive in the memory of the TIU unless that data is of the proper level. Thus we may be forced to buffer the destination and source fields of an incoming packet within the security processor until we can be sure of the security level, instead of simply passing the fields into TIU memory "on the fly". Also, if a collision has occurred, which will always be detected during reading of the destination or source fields, we may not want to have the partially-read data in memory. Finally, there is the possibility that a packet with a frame-check error will be fully read into memory before the error is detected. There would then be a slight chance that the security level field was corrupted and that the packet should not have been accepted. The probability of this happening is extremely low, since untrusted TIU software cannot force a frame-check error to occur at will, and even if one should occur, there is little likelihood that the error will be such that both the destination and security level have precisely the required values. Furthermore, since the error is a random event, there is no way for malicious software in the TIU to modulate this type of error for covert communication. In any case, full-frame buffering in the security processor could be implemented.

The reason only single-level subscribers can be supported with the architecture outlined here is that nothing outside of the TIU CSMA/CD interface and security processor (e.g., terminals, hosts, TIU software) is trusted to maintain separation of data of different levels. This is the most common situation in today's environments.

Variable-level TIUs

A variable-level TIU is the same as a single-level TIU, except that an operator can change the level of the TIU from time to time. This is accomplished via some manual interface to the security processor (e.g., rotary switch). Procedural controls should insure that the host or terminal is appropriately sanitized when the level of the TIU is lowered. This sanitization can be accomplished automatically using a number of techniques. The variable-level TIU might also interface to special-purpose keys on a user's terminal keyboard—keys that are electrically linked to the security processor.

A more complex variable-level TIU for a terminal might allow the operator to communicate the change of security level via the normal keyboard and screen of his terminal. This would entail, however, significantly more complex mechanisms that must be trusted. In figure 4, for example, all of the software in the TIU would have to be trusted, since that code processes keyboard input before it is seen by the security processor. Sophisticated, but well-understood, techniques to implement a logical "trusted communications path" from operator to security processor would have to be employed. Of course for a host that undergoes periods processing to handle classified data of different levels at different times, only a manual interface to the TIU security processor would be appropriate.

Multilevel TIUs

The multilevel TIU for a host or terminal is likely to contain fully trusted software. The security processor in such a TIU would only be able to limit communications to the range of levels at which the host or terminal is authorized to

*This separation is similar to the "red/black" separation employed with crypto hardware. In fact, when the "untrusted" side is unclassified, many of the conventional physical red/black separation requirements would have to be adhered to.

operate. The rest of the TIU would have to be trusted to properly identify the security level of the data to the host within that range, so that the host (which is trusted) can make the correct decisions to provide the necessary protection of the multilevel data. In terms of total functionality and complexity, a multilevel TIU is the same as a single-level TIU. The only difference is in the degree of trust given to the software and hardware in the TIU that is not in the security processor. Thus, the difficulty of building a multilevel TIU over that of a single-level TIU is dependent on software engineering techniques (e.g., verification) rather than on inherent complexity.

BRIDGES

Bridges operate strictly at the CSMA/CD protocol level. A bridge always connects exactly two subnetworks (a simplifying requirement). Its job is to pick packets from one subnetwork, check their destinations and security levels, and send them to the other subnetwork. To prevent congestion, bridges must operate at a speed fast enough to handle a reasonable load of traffic from one subnetwork to another, although multiple bridges could be used to help.

Figure 5 shows the logical operation of the bridge. Note that the destination check is made as the packet is read in from the network before it is buffered, exactly the same as is done by the CSMA/CD interface in the TIU. In the case of bridge, however, more than a single destination must be checked. The set of destinations accepted are stored in a fixed routing table that can be quickly scanned at network speeds. A two-level hierarchical addressing structure is employed to simplify this lookup and reduce the size of the table (see figure 3). Each bridge knows exactly which subnetworks it is responsible for. Thus, packets are buffered in the bridge only if they are definitely addressed to another subnetwork to which the bridge is logically linked.

The security processor in the bridge makes the appropriate security checks on the buffered packets, based only on the levels of the two subnetworks immediately adjacent to the bridge. The bridge does not check final TIU or subnetwork destinations, nor does it verify that the level of the packet is correct with respect to the source. It only checks that the received packet is labelled within the range of levels of the subnetwork from which the packet is arriving, and that that level is within the range of levels of the next subnetwork.

Outgoing packets are handled by the CSMA/CD interface from output buffers in a manner identical to outgoing packets in a TIU.

Because the bridge contains no protocol software outside of the CSMA/CD interface, and the only function it implements is a simple security level check, we see the bridge as simple enough to be fully trusted to perform its job correctly and fast enough to handle a considerable load. The

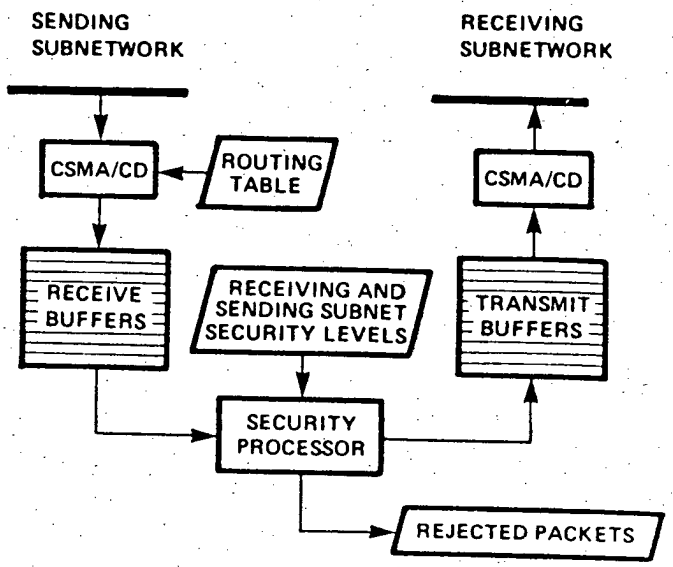

Figure 5. Bridge

buffers in the bridges smooth out temporary overloads. It is not expected that the bridges would be a bottleneck in the overall system except in times of heavy continuous traffic. For this reason the approach is recommended in environments where high traffic density is not expected in the near term. To better deal with greater traffic loads, and to provide more flexibility in addressing and routing, the bridges should provide some form of congestion control as might be implemented in a higher protocol layer such as the DoD Internet Protocol (IP). Such a change should be implemented as a future enhancement (along with corresponding changes to the TIUs to use IP) as it would significantly complicate the amount of trusted software in the bridges and TIUs.

Adding certain key aspects of IP to the bridges, in place of some portions of the CSMA/CD protocol, would allow for internet routing among the local subnetworks and through gateways to wide area networks. This means that the subnetworks would be more like separate networks and the bridges would be more like gateways. IP in the bridges and TIUs would also allow incorporation of the security level field into the header where it is specified as an option in IP, rather than usurping part of the data field of CSMA/CD. Finally, IP would allow a primitive form of congestion control--a bridge could return a control packet to a sender to turn off further transmissions due to overload in the bridge or adjoining subnetwork.

While there are many advantages to putting IP in the bridge, we do not feel at this point that the extra complexity in the TIUs and bridges is desirable in an initial configuration considering the need to trust the software. It may be very likely that an initial installation of a secure LAN would indeed require IP in the TIUs for internetworking [Skelton80], but that IP implementation would reside in the untrusted portion of the TIU

and would not be interpreted by the bridges. A smooth transition to installation of IP in the bridges may involve, in part, verifying the existing IP software in the TIUs for multilevel operation.

The two half-bridges shown in figure 2 comprise a special form of bridge required when encryption is necessary between two classified subnetworks. This will be discussed further in the following section.

ENCRYPTION

In Figure 2 several locations are shown where encryption is required. At the top an encrypted line is shown between the confidential host and its TIU that resides in the secret environment. This line would probably employ conventional bit-serial link encryption at the appropriate speed (ignore, at this point the dubious need for encryption on a confidential line).

Another encrypted line is shown on the right side between the Top Secret subnetworks. Encryption here is employed directly between the media of the two subnetworks, without the use of a bridge. This is intended to illustrate encryption of the LAN medium where a cable may pass, for example, between two Top Secret protected buildings. We have not studied the problem of encrypting the LAN medium directly, and what affect it might have on the physical and CSMA/CD protocols. However, our design is not dependent on the ability to encrypt such media, as the subnetworks could be separate and bridges could be used instead.

Near the center of the figure are shown two "split-bridges"—one in the Top Secret environment and the other on the Secret subnetwork. Each half of the bridge communicates with its subnetwork directly using the straightforward CSMA/CD protocol. The two halves of the bridge communicate via a serial line that can be encrypted using conventional means. The functionality of the bridge is allocated between the two halves according to the security requirements. For example, the security processor, which checks that incoming packets from the high side only go to the low side if they have the appropriate security level, must be located on the high side. Buffering for transmission to the low side, and part of the IP protocol handling, if implemented, could be on the low side. Note that the split bridge concept, while introduced here to deal with the encryption problem, is a general solution where two subnetworks cannot be brought into close physical proximity.

The split-bridge is considerably more complex than a single bridge, and it would not be needed in cases where encryption is not required and the media of the two subnetworks could be brought close together. A bridge between a classified subnetwork to an unclassified subnetwork would not have to be split.

CONCLUSION

This secure local area network architecture presented here is one of several means by which multilevel data on a local network can be protected. This design stresses very near-term availability, and as such makes maximum use of well-understood concepts, existing protocols, and off-the-shelf hardware. In order to assist certification for multilevel operation, a minimal amount of trusted software or firmware is required. The TIU design provides a basic trusted multilevel service that allows for implementing in additional TIU software a wide range of applications. This additional software need not be certified or verified. Finally, the architecture is designed to be implemented in phases, providing an initial capability that integrates well into existing operations, and is then upgradable to full service as the need arises.

REFERENCES

Clark78 — Clark, D. D., Pogran, K. T., and Reed, D. P., "An Introduction to Local Area Networks," **Proc. IEEE**, Vol. 66, No. 11, pp. 1497-1517, November 1978.

DoD72 — Department of Defense Directive, DoD 5200.28 "Security Requirements for Automatic Data Processing (ADP) Systems," December 18, 1972.

Ethernet80 — The Ethernet: A Local Area Network Specification, Version 1.0, DEC, INTEL, XEROX, September 30, 1980.

IEEE81 — **Local Network Standards Committee, A Status Report**, Draft B, IEEE Computer Society, October 19, 1981.

ISO81 — ISO/TC97/SC16, "Data Processing—Open Systems Interconnection—Basic Reference Model," **Computer Networks**, Vol. 5, 1981, pp. 81-118.

Nibaldi79 — Nibaldi, G. H., "Specification of a Trusted Computing Base (TCB)," M79-228, The MITRE Corporation, Bedford, MA, November 30, 1979.

Postel81a — Postel, J. (ed.), "DoD Standard Internet Protocol," Defense Advanced Research Projects Agency, 1981.

Postel81b — Postel, J. (ed.), "DoD Standard Transmission Control Protocol," Defense Advanced Research Projects Agency, 1981.

Skelton80 — Skelton, A. P., Nabielsky, J., and Holmgren, S. F., "FY80 Final Report: Cable Bus Application in Command Centers," MTR-80W00319, The MITRE Corporation, McLean, VA, December 1980.