Secure Communication in Internet Environments: A Hierarchical Key Management Scheme for End-To-End Encryption

WEN-PAI LU, MEMBER, IEEE, AND MALUR K. SUNDARESHAN, MEMBER, IEEE

Abstract—**The problem of designing key management schemes for establishing end-to-end encrypted sessions between source-destination pairs when the source and the destination are on different networks interconnected via gateways and intermediate networks is considered. In such an internet environment, the key management problem attains a high degree of complexity due to the differences in the key distribution systems used in the constituent networks and the infeasibility of effecting extensive hardware and software changes to the existing networks. In this paper, a hierarchical approach for key management is presented which utilizes the existing network specific protocols at the lower levels and protocols between authentication servers and/or control centers of different networks at the higher levels. Details of this approach are discussed for specific illustrative scenarios to demonstrate the implementational simplicity. A formal verification of the security of the resulting system in the sense of protecting the privacy of privileged information is also conducted by an axiomatic procedure utilizing certain combinatory logic principles. This approach is general and can be used for verifying the security of other existing key management schemes.**

I. Introduction

THE level of security provided by an encryption-based scheme for secure communication across a data network is highly dependent on the security of the keys used for the encryption and decryption of data. To ensure a high security level, these keys should be changed often; ideally, an updating on a session-by-session basis, i.e., a unique "session key" generated for each new session and discarded at the end of the session, is highly desirable. The feasibility of such an updating, particularly when end-to-end encryption is contemplated, depends on the existence of a key management mechanism that facilitates, at the initiation of each new session, the generation of a session key and its distribution to the two end communicants. Furthermore, it is desirable that this transfer be made on the existing communication channels, which in turn demands the highest level of security during such a transfer. Consequently, key management is perhaps more important to the working security of a network than the mathematical structure of the encryption algorithm itself, since an inefficient key transfer between the end communicants can make the entire scheme worthless regardless of how complex the encryption itself is.

The problem of designing key management protocols for a communication network implementing end-to-end encryption has attracted considerable attention in the recent times and some early contributions are due to Ehrsam *et al.*, [1], Matyas and Meyer [2] and Needham and Schroeder [3]. In [1] and [2] are described specific procedures for the generation of a session key (SK), its transfer between the two nodes involved in the session while encrypted under a unique master key (MK) used for this exchange, and the secure storage of MK and SK at the nodes. In [3], the use of a centralized facility, called an authentication server (AS), that performs not only session key generation and its secure distribution, but also authenticates the connections, is advocated. Specific protocols for the establishment of authenticated connections in the case of both conventional and public-key encryption algorithms are also presented. More recently, improvements to the Needham-Schroeder protocols to prevent security breaches when exchange keys used in the distribution of SK are compromised, have been developed using the notion of timestamps by Denning and Sacco [4] and alternately, using event markers by Bauer *et al.*, [5]. Some of these key management protocols have also been implemented in certain operational networks [6].

The key management problem attains a greater degree of complexity in internet environments where communication units located on different networks desire secure exchanges with one another. To describe precisely the additional complexities posed, consider a typical example of an internet system that includes two secure local networks LN1 and LN2 interconnected directly through a Gateway [7] or possibly via two gateways and an intermediate long distance network [8]. The long distance network may or may not support security services; that is, capabilities for encryption and and decryption of data at its nodes before transmitting out may or may not be present. The internet system may also include several other networks, some of which may not have security features. Let us assume that the two secure networks LN1 and LN2 process the same encryption algorithm and have in place appropriate key management mechanisms to support end-to-end encryption between the nodes of the same network. In such an environment, when a secure session with end-to-end encryption across the internet is desired, several additional problems are to be taken into consideration. First, the demands on the security mechanism and the verification of security will be higher due to the greater vulnerability to attacks caused by the increased possibilities for intercepting messages. Second, and more seriously, the key management protocols in the individual networks may not be uniform and the establishment of both internet and intranet sessions with the same protocol may be impossible.

While attempts to overcome the first problem can be made by tighter security measures (access control, physical measures, etc.), the second problem is more difficult to handle due to implementational constraints (and some associated addressing problems as will be described later). Note that a solution, at least simple conceptually, is possible if the commands and the associated software for key management in the individual networks (both at the network nodes and at the key generation and distribution faility) and the gateway processing functions can be altered as extensively as desired. Such a solution,

Paper approved by the Editor for Wide Area Networks of the IEEE Communications Society. Manuscript received April 28, 1986; revised July 29, 1987 and April 12, 1988. This paper was presented at the 1986 IEEE Symposium on Security, Oakland, CA, April 1986.

W.-P. Lu is with AT&T Bell Laboratories, Holmdel, NJ 07733.

M. K. Sundareshan is with the Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85712.

IEEE Log Number 8930073.

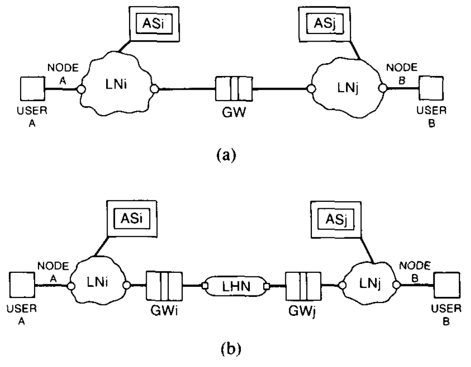


Fig. 1. (a) Scenario 1: Local networks directly connected by a gateway. (b) Scenario 2: Local networks connected via a long-haul network.

however, is quite unrealistic to implement since networks and network components are standardized products and any requirement of modifying the software of already installed networks is rather unattractive. Consideration of a typical illustrative example of the internet environment created by the multitude of different LAN's existing in different military installations across the country being interconnected by the defense data network and appropriate gateways [8] will at once make this point very clear.

The problem of interest in establishing key management schemes for internet secure sessions with end-to-end encryption is then the following. How to develop the required protocols for key generation and key distribution by utilizing the existing intranet protocols within the constituent networks with minimal changes to the network software? In this paper, a hierarchical key management scheme is presented to answer this question. The hierarchy is comprised of the intranet protocols at the lower levels and protocols between control centers (key generation facilities of individual networks, control centers for clusters of networks, etc.) at the higher levels. An illustrative application to a specific internet scenario is outlined. An axiomatic approach is employed for a formal verification of security afforded by the key management scheme.

II. End-to-End Secure Communication in Internet Environments

A. Specific Problem of Interest

In order to precisely describe the approach that is used and the details of the protocols needed, two specific interconnection scenarios will be considered. To ensure a sufficiently high degree of generality in the problems considered, a mix of both LAN's and long-haul networks in the internet environment will be assumed.

Consider the scenario depicted in Fig. 1(a) where a communication node A on a local network LN_i is desirous of having a secure session with a communication node B on a remote local network LN_j. We shall assume that LN_i and LN_j are secure networks, i.e., an appropriate mechanism to ensure end-to-end secure communication between the nodes of the same network is in place in each network. We shall further assume that a common encryption algorithm of the conventional-key type (such as DES) is used and the encryption/decryption of messages are performed at the session layer of the OSI protocol hierarchy [9]. Each network supports the use of a unique session key SK for each new session, which is generated by an authentication server at the beginning of the session and is distributed to the two end communicants following an established protocol (for instance, any protocol of the type described in [1], [3]-[5]). The key generation facility at the AS consists of a random number generator that outputs a key of fixed length based on the current state value [2]. The specific protocol used for key distribution in each network may be identical or may be dissimilar. The internet communication path from node A on LN_i to node B on LN_j includes an appropriate gateway GW connecting LN_i to LN_j.

The scenario depicted in Fig. 1(b) describes a similar connection requirement with the difference that the communication path from node A on LN_i to node B on LN_j includes an appropriate local gateway GW_i connecting LN_i with a long-haul network LHN and a remote gateway GW_j connecting LN_j with LHN. As before, LN_i and LN_j are assumed to be secure networks. LHN may be a secure network or an unsecure network; the presence or absence of security on LHN is not of particular concern to our present work due to our interest in end-to-end encryption between the source and the destination (,viz. node A and node B).

The problem of interest is to design a suitable key management scheme for generating and distributing session keys at the initiation of eah new session for end-to-end secure communication between node A and node B. As mentioned in the Introduction, in the solution to such a problem, proper attention should be given to the constraints imposed by the practical implementation of the required protocols, and consequently, we shall stipulate that no major changes to the existing key distribution protocols and the command structures of the constituent networks be demanded. It should be appreciated that a tradeoff exists between the degree of relaxing the above requirement and the conceptual complexity of the solution to the problem.

B. Typical Key Management in a Single Network

For a precise description of the solutions to such an internet key management problem, it is necessary to specify the details of key generation and distribution within a single network. Several schemes have been proposed in the literature for the management of session keys within a network. These schemes typically assume the existence of an authentication server (AS) which will authenticate the two end communicants (nodes A and B) and generate a *unique session key* SK for each new session and of a key distribution protocol for transferring the SK in an encrypted form to the nodes A and B on the existing communication channels. To permit a secure transfer of SK, each node is equipped with a unique master key MK which will also be known to AS.

For a concise description of the exchanges involved in such a transfer, let us consider the two illustrative schemes shown in Fig. 2(a) and 2(b). Let $(x)K$ denote the message x encrypted under key K. Then the scheme described in Fig. 2(a), originally given by Bauer *et al.*, [5] involves the following steps when node A initiates an attempt to open a secure session with node B. It will be assumed that the master kay of A, MK_a, is available only at node A and at AS. Similarly, the master key of B, MK_b, is available only at node B and at AS.

$$\text{Step 1) } A \rightarrow B : \{A, (\mathrm{EM}_a)_{\mathrm{MK}_u}\} \tag{1}$$

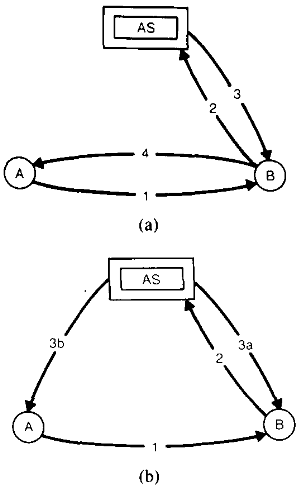
where A is the identity of node A and EM_a is an event marker, which is an identifier used by node A to establish a one-to-one correspondence between his attempt to initiate the secure session and the resulting session key SK.

$$\text{Step 2) } B \rightarrow \mathrm{AS} : \{(A, (\mathrm{EM}_a)_{\mathrm{MK}_a}, B, (\mathrm{EM}_b)_{\mathrm{MK}_b}\} \tag{2}$$

where the identity of node B and a new event marker EM_b are added to the message by node B.

$$\text{Step 3) } \mathrm{AS} \rightarrow B : \{(\mathrm{SK}, A, \mathrm{EM}_b)_{\mathrm{MK}_b}, (\mathrm{SK}, B, \mathrm{EM}_a)_{\mathrm{MK}_a}\} \tag{3}$$

where SK is the session key generated by AS.



(a)

(b)

Fig. 2. (a) A typical key management scheme within a single network. (b) An alternate key management scheme.

Step 4) B decrypts the first part of the message under his key MK_b to obtain SK and transmits the remainder of the message to A, i.e.,

$$B \to A : \{(SK, B, EM_a)_{MK_a}\}. \tag{4}$$

Now A decrypts this message and obtains SK.

In Fig. 2(b) is shown a simple modification of the above scheme where Steps 3) and 4) are replaced by the following.

$$\text{Step 3a) } AS \to B : \{(SK, A, EM_b)_{MK_b}\}$$

and

$$\text{Step 3b) } AS \to A : \{(SK, B, EM_a)_{MK_a}\}, \tag{5}$$

which may be executed in parallel.

C. Key Management in the Internet Environment

For the scenario described in Fig. 1(a), let AS_i and AS_j denote authentication servers of the two networks LN_i and LN_j, respectively. We shall assume that the master key of each node of LN_i is stored at that node and also at AS_i in a master key table. Similarly, the master key of each node in LN_j is stored at that node and also at AS_j in a corresponding table. The master key tables at AS_i and AS_j are stored in an encrypted form under appropriate storage keys. AS_i and AS_j support all the functions described for AS in the previous section.

Since our primary objective in establishing a key management scheme for the internet is to keep the required changes to the local network protocols to a minimum, let us restrict the needed changes to only the authentication servers and the gateway, and demand that no changes (hardware or software) be made to the nodes. This is not objectionable due to the presence of only one authentication server per network and one gateway per pair of networks, while the number of nodes will be huge. Also, since the nodes are distributed across the network, implementing the required changes when a network in operation is brought into the internet system will be more tedious and costly. Furthermore, the authentication server will usually be a part of a bigger network control center (NCC) which can accommodate implementing the required changes in a simple manner.

To describe the present key management scheme, let us begin by listing the required changes to the authentication servers. The generation and distribution of a session key will require a handshake between AS_i and AS_j. For this exchange, which is a protocol at a higher hierarchical level, let us assume the presence of an inter-AS exchange key EK_{ij}, which is unique to each pair of authentication servers AS_i and As_j in the internet environment (i.e., $EK_{ij} = EK_{ji} \; \forall \; i, j \ni i \neq j$.) In particular, EK_{ij} is securely stored at AS_i and AS_j under the storage keys being used at each AS.

In order to retain the nodal protocols unchanged, the process of secure session establishment should begin with the source node A on LN_i to address the destination node B as in Step 1) of the earlier described protocol. However, since the destination is on a different network now, there could in general be problems in the source directly addressing the remote destination, due to common numbering of the nodes in the two networks or the address sizes and formats being different, etc. Solutions to such internet addressing problems can be obtained in different forms, a typical one being a hierarchical addressing scheme reported in [10]. In this approach, an interactive session is established between the source and a directory service at the NCC of the network on which the source resides for obtaining the internet address of the destination from a logical name supplied by the source. The exchanges involved are briefly summarized in the following sequence of steps.

Step 1) $A \to$ NCC: "Help"

Step 2) NCC $\to A$: Destination?

Step 3) $A \to$ NCC: "Node B on Network LN_j"

Step 4) NCC identifies the appropriate gateway GW from the directory, supplies the address of the destination B to GW and leaves the source A and GW in session using the network specific protocols.

More details on the procedure and some illustrative examples of applying the scheme to an internet environment created by interconnecting different types of LAN's via the defense data network and gateways can be found in [10]. It is to be noted that this approach does not require any changes to the communication nodes, all the modifications being localized to the NCC and GW.

To overcome the complexities posed by the internet addressing problem, let us expand the authentication server for each network to include a directory service to support the hierarchical addressing scheme described above. With this, the key management scheme in the internet system can be divided into two major parts: 1) addressing the destination and 2) session key generation and distribution. A concise description of the steps involved when node A on LN_i initiates an attempt to open a secure session with node B on LN_j will now be given.

1) Basic Internet Protocol:

$$\text{Step 1) } A \to AS_i : \{(A, B, LN_j)_{MK_a}\} \tag{6}$$

which summarizes the interactive exchanges described earlier.

$$\text{Step 2) } AS_i \to AS_j : \{(A, B, EM_{asi})_{EK_{ij}}\} \tag{7}$$

where EM_{asi} is an event marker generated by AS_i for authentication purposes (as described earlier). This handshake takes place via GW.

Step 3) AS_j generates session key SK and sends the following message to AS_i

$$AS_j \to AS_i : \{(SK, A, EM_{asi})_{EK_{ij}}\}. \tag{8}$$

AS_i obtains SK by decrypting this message.

The remaining steps perform the distribution of SK by AS_i to A and by AS_j to B using the existing network specific protocols, which can take place in parallel.

Step 4) AS_i instructs A to make a call [1] to GW to transmit

[1] In several networks a remote call procedure exists that permits the NCC to leave two nodes on the network in a session.

the following message:

$$A \rightarrow GW : \{A, (\mathrm{EM}_a)_{\mathrm{MK}_a}\}. \tag{9}$$

AS_j instructs B to make a call to GW to transmit the following message:

$$B \rightarrow GW : \{B, (\mathrm{EM}_b)_{\mathrm{MK}_b}\}. \tag{10}$$

Step 5)

$$GW \rightarrow \mathrm{AS}_i : \{A, (\mathrm{EM}_a)_{\mathrm{MK}_a}, GW, \mathrm{EM}_{gw}\}$$

$$GW \rightarrow \mathrm{AS}_j : \{B, (\mathrm{EM}_b)_{\mathrm{MK}_b}, GW, \mathrm{EM}_{gw}\}. \tag{11}$$

Step 6)

$$\mathrm{AS}_i \rightarrow GW : \{(\mathrm{SK}, GW, \mathrm{EM}_a)_{\mathrm{MK}_a}, (A, \mathrm{EM}_{gw})\}$$

$$\mathrm{AS}_j \rightarrow GW : \{(\mathrm{SK}, GW, \mathrm{EM}_b)_{\mathrm{MK}_b}, (A, \mathrm{EM}_{gw})\}. \tag{12}$$

Step 7)

$$GW \rightarrow A : \{(\mathrm{SK}, GW, \mathrm{EM}_a)_{\mathrm{MK}_a}\}$$

$$GW \rightarrow B : \{(\mathrm{SK}, GW, \mathrm{EM}_b)_{\mathrm{MK}_b}\}. \tag{13}$$

Both A and B decrypt the received messages and load SK into appropriate registers. Since the exchanges in Steps 4)–7) take place in parallel in two different networks, it is necessary to ensure that SK is available at both A and B before the users at these nodes commence sending messages. Such a synchronization can be simply achieved by GW signaling the call completion status to both A and B simultaneously after receiving acknowledgment at the end of Step 7).

Alternately, if the key management protocol within the individual networks conforms to Steps 3a) and 3b) instead of Steps 3) and 4) in Section II-B., we could replace Steps 6) and 7) above by the following:

Step 6a)

$$\mathrm{AS}_i \rightarrow A : \{(\mathrm{SK}, GW, \mathrm{EM}_a)_{\mathrm{MK}_a}\}$$

Step 6b)

$$\mathrm{AS}_j \rightarrow B : \{(\mathrm{SK}, GW, \mathrm{EM}_b)_{\mathrm{MK}_b}\}$$

which may be executed in parallel. In this case, synchronization between A and B can be achieved by having AS_i and AS_j send messages to GW who in turn will signal the call completion status to both A and B.

It may be noted that the parts of the messages to be retained by GW in Step 6) are not encrypted, since we do not propose that GW be equipped with encryption/decryption capability. This may be useful since the gateway is a point of high vulnerability in internet communications and the nonavailability at this place of a device processing the encryption algorithm being used is rather useful. However, if GW possesses encryption/decryption capabilities like all other nodes in the networks to which it is connected, the second part of the messages in Step 6) could be encrypted under a master key of GW to retain exact correspondence with the protocol for the single network case outlined in Section II-B.

The detailed exchanges given in Steps 4)–7) for key distribution within the two local networks are only illustrative and demonstrate how a typical protocol outlined in Section II-B can be adapted for this purpose. As is evident, these may be replaced by any other protocols in place in the networks (these need not be identical in the two networks). One such protocol that has received considerable attention recently is the ANSI X 9.17 standard operating in the key distribution center environment [16]. This underscores the generality of the present approach to internet key management which can easily adapt any specific intranet protocol of choice for the distribution of encryption keys within the individual networks.

For key management in Scenario 2 depicted in Fig. 1(b), an extension of the above protocol can be directly made, noting that two gateways GW_i and GW_j are involved in the key distribution and hence a two-way handshake between the two gateways to synchronize call completion signaling will be required. Some modifications to the implementation of the key management scheme could be envisaged due to the large number of local networks included in the internet system being interconnected by the long-haul network. Note that large databases to store the inter-AS exchange keys of all nodes in every network and addresess become necessary at the authentication server of each network. To avoid maintaining such large and duplicate databases, these tables can be set up at a single centralized location, perhaps at the control center of the long-haul network, LHCC. Maintaining such a centralized database would also be more attractive when changes to the structure of the internet system due to the addition and deletion of networks are to be frequently expected.

Using a centralized database however adds another level of hierarchy to the exchanges in setting up a secure session. In particular, Step 2) in the above protocol needs to be replaced by the following steps.

Step 2a)

$$\mathrm{AS}_i \rightarrow \mathrm{LHCC} : \{(A, B, \mathrm{LN}_j, \mathrm{EM}_{asi})_{\mathrm{EK}_{Ci}}\}. \tag{14}$$

Step 2b)

$$\mathrm{LHCC} \rightarrow \mathrm{AS}_i : \{(\mathrm{EK}_{ij}, \mathrm{AS}_j, B, \mathrm{EM}_{asi})_{\mathrm{EK}_{Ci}}\}$$

$$\mathrm{LHCC} \rightarrow \mathrm{AS}_j : \{(\mathrm{EK}_{ij}, \mathrm{AS}_j, A, \mathrm{EM}_{asi})_{\mathrm{EK}_{Ci}}\}. \tag{15}$$

Step 2c)

$$\mathrm{AS}_i \rightarrow \mathrm{AS}_j : \{(A, B, \mathrm{EM}_{asi})_{\mathrm{EK}_{ij}}\}. \tag{16}$$

In the above exchanges, AS_i and AS_j provide the identities of the two authentication servers to each other. The event marker EM_{asi} reeived by AS_j from LHCC in Step 2b) and from AS_i in Step 2c) enables it to associate EK_{ij} with the correct session initiation attempts. For secure transmission of EK_{ij} to AS_i and AS_j, the messages in Step 2b) are encrypted under exchange keys EK_{Ci} and EK_{Cj} which will be used by AS_i and AS_j, respectively, for all secure exchanges with LHCC. To make these exchanges possible, we shall assume that EK_{Ci} is stored at AS_i and at LHCC, and EK_{Cj} is stored at AS_j and at LHCC.

Despite the advantages of a centralized database as noted above, maintaining duplicate databases at all authentication servers may be attractive to prevent excessive traffic to LHCC (limiting this traffic would enable LHCC to perform other control functions) when establishment of internetwork sessions becomes frequently necessary. The problems associated with performing additions to the exchange key tables when new networks are brought into the internet system can be overcome by a simple procedure. An authentication server requests the LHCC for the exchange key for his counterpart in the new network only at the first attempt for a session, after which this key will be added to his table in order to facilitate all future exchanges directly.

D. Discussion of the Key Management Protocol

A) The key management scheme described in the previous section is hierarchical in nature and involves distinct protocols at different levels of the hierarchy. At the lower levels, we have the protocols between the network nodes (gateway is considered as a node on each network it interconnects) and between a node and the AS, while at higher levels there are the inter-AS protocols and the protocols between an AS and the LHCC. The hierarchy can be extended further to distribute the functions of the LHCC, particularly when the internet system includes a very large number of local networks making a single control center infeasible due to the large databases and excessive overhead traffic converging to the control center. In such cases, a higher degree of efficiency can be achieved by dividing the internet system into several clusters of local networks and defining a regional control center (RCC) for each cluster [14]. Each RCC stores the inter-AS exchange keys of every pair of authentication servers of the networks within the cluster and also the addresses of the nodes in these networks.

The exchanges required in establishing a secure session between two nodes on two different networks which are part of two different clusters will now require an inter-cluster protocol between the two RCC's. This exchange, for security purposes, will be encrypted under an inter-RCC exchange key EK_{RCij} (for exchange between RCC_i and RCC_j) and hence we will assume the presence of an inter-RCC exchange key table at each RCC. The addressing and session key exchange will not involve any major modifications to the basic protocol, except that Step 2) is now replaced by the following steps:

$$\text{Step 2a)}\ \mathrm{AS}_i \rightarrow \mathrm{RCC}_i : \{(A, B, \mathrm{LN}_j, \mathrm{EM}_{asi})_{\mathrm{EK}_{Ci}}\} \tag{17}$$

$$\text{Step 2b)}\ \mathrm{RCC}_i \rightarrow \mathrm{RCC}_j : \{(\mathrm{AS}_i, A, B, \mathrm{LN}_j, \mathrm{EM}_{asi})_{\mathrm{EK}_{\mathrm{RC}\ ij}}\} \tag{18}$$

$$\text{Step 2c)}\ \mathrm{RCC}_j \rightarrow \mathrm{RCC}_i : \{(\mathrm{EK}_{ij}, \mathrm{AS}_i, B, \mathrm{EM}_{asi})_{\mathrm{EK}_{\mathrm{RC}\ ij}}\} \tag{19}$$

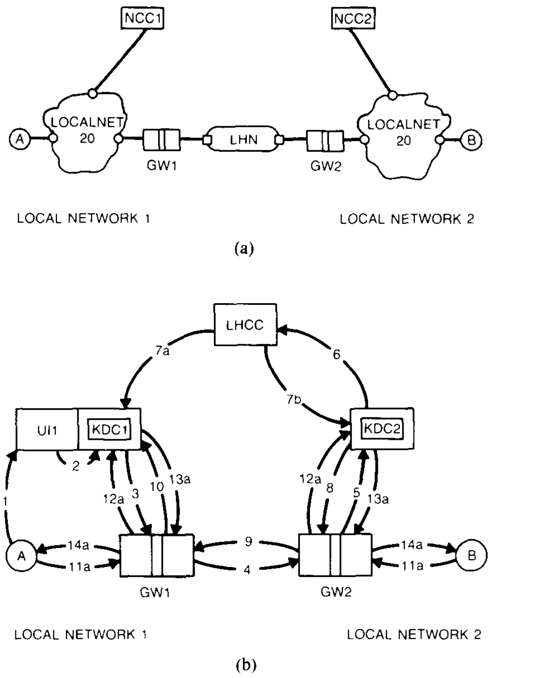
Step 2d)

$$\begin{aligned} &\mathrm{RCC}_i \rightarrow \mathrm{AS}_i : \{(\mathrm{EK}_{ij}, \mathrm{AS}_j, B, \mathrm{EM}_{asi})_{\mathrm{EK}_{Ci}}\} \\ &\mathrm{RCC}_j \rightarrow \mathrm{AS}_j : \{(\mathrm{EK}_{ij}, \mathrm{AS}_j, A, \mathrm{EM}_{asi})_{\mathrm{EK}_{Cj}}\} \end{aligned} \tag{20}$$

$$\text{Step 2e)}\ \mathrm{AS}_i \rightarrow \mathrm{AS}_j : \{(A, B, \mathrm{EM}_{asi})_{\mathrm{EK}_{ij}}\}. \tag{21}$$

B) A modification of the presently developed protocol can be obtained by using public key encryption [11], [12] for the exchange of SK (which, as before, will be used for encrypting actual messages under a conventional key scheme). For implementing this, each authentication server is equipped with directories of the public keys of all nodes in the network it serves and of the public keys of all other authentication servers in the internet system. For brevity, details of this modification will be omitted here; the required changes are, however, straightforward and follow similar modifications discussed in the literature [3]-[5].

C) The security afforded by the present hierarchical scheme depends naturally on the security of the keys stored at the various levels, which may be ensured either by using appropriate hardware or by encryption under appropriate storage keys and implementing effective authentication checks. Evidently, the higher the level at which security breaches resulting in the compromise of keys takes place, the more damaging the effect will be on the overall security of information transfer and the more complex the recovery process will become. For instance, if the attack is at the lowest level, i.e., at a communication node, the worst case is that the master key MK of that node and a session key SK of a session in which the node is currently engaged will be known to the intruder. The effect thus is limited to the node and the recovery is rather simple in requiring MK to be updated and a new session key to be issued. On the other hand, if the attack is at a higher level, say at an authentication server, the worst case will be that the master keys of all the nodes in that network and the inter-AS exchange keys will be exposed. The damage consequently is more widespread and the recovery process becomes more complex.



(a)

(b)

Fig. 3. (a) Specific application scenario: two LocalNet 20 networks connected by a long-haul network. (b) Protocol details for proposed scheme.

III. Development of a Specific Key Management Protocol for an Illustrative Internet Environment

In order to precisely describe how the present hierarchical approach to key management can be employed in practice to develop specific protocols, let us consider an illustrative application scenario depicted in Fig. 3(a) with the two local networks being Sytek LocalNet 20 networks [2] interconnected via a Long-Haul Network such as the Defense Data Network (DDN). We shall begin by reviewing some essential features of the LocalNet 20 network.

LocalNet 20 is a broadband LAN that uses CSMA/CD media access. The basic packet switching unit is called a packet communication unit (PCU) which provides user devices with access to the cable network. Each PCU on the network is identified by a unique 16-bit Unit ID.

The network control center (NCC) of the LocalNet 20 provides a variety of services [13] for network management and operation. The NCC software provides a UNIX-based multitasking environment and a structured file system and consists of several service programs of which the "user interface" (UI) is of particular interest. The UI provides "Name Service" to assist in addressing of destinations by performing a translation of symbolic names into LocalNet addresses for session establishment and access control. The UI operates on a "destination file" for name service to accept a symbolic name of the destination and give the user the corresponding unit ID.

Secure sessions between two PCU's on the same local network are established by a key distribution center (KDC) within the NCC. The steps which are involved in the generation and distribution of a session key SK when a user on PCU*A* initiates a secure session with a user on PCU*B* by issuing a *SCALL* command follow the protocol outlined in Section II-B. If the destination address is not known, then the

[2] LocalNet 20 is a trademark of Sytek, Inc.

establishment of the secure session can take place with the help of the name service provided by the UI program in the NCC.

A. Key Management Protocol

To follow the general approach outlined earlier in Section II, let us assume the presence of a control center for the long-haul network (LHCC) whose functions are expanded to include the protocols to support exchanges between LHCC and each KDC of the local networks present in the internet system. Specifically, we shall require that LHCC can generate, on request from KDC_i, a unique exchange key EK_{ij} that will be used by KDC_i for secure exchanges with KDC_j during the specific session establishment attempt. For secure transmission of EK_{ij} from LHCC to KDC_i, we shall assume the availability of a master key MK_{Ci} unique to KDC_i which will be securely stored only at KDC_i and at LHCC in a master key table. Thus KDC_i will use the master key MK_{Ci} for secure exchanges with LHCC, the exchange key EK_{ij} for secure exchanges with KDC_j in the internet system and the master key MK_a for secure exchanges with PCUA on its network.

Since the protocol for opening a secure session at each PCU should remain unchanged, the SCALL command should be initiated by PCUA on the LocalNet 1 to address PCUB on the LocalNet 2. Since the intended destination is on a different network, one immediately encounters the problem of addressing, to which the hierarchical addressing scheme developed by Sundareshan and Muralidhar [10] can be exployed. A description of the application of this approach to the present problem can be found in [15].

The generation and distribution of the session key for the session between PCUA and PCUB will proceed along the following steps [see Fig. 3(b)].

Step 1) User on PCUA first opens a session with UI_1 to request the address of PCUB.

Step 2) UI_1 identifies the appropriate local and remote gateways ($GW1$ and $GW2$) and the KDC at the remote network (KDC2) from its directory. It then sends the following message to KDC1:

$$\mathrm{UI}_1 \rightarrow \mathrm{KDC1} : \{ss, A, B, GW1, GW2, \mathrm{KDC2}\}.$$

In this message, the first term is an identifier for a secure session while the remainder terms are the identifiers of the PCU's, gateways and the remote KDC that will be involved in the session establishment.

Step 3) KDC1 generates two event markers; $\mathrm{EM}_{\mathrm{KDC1}}$ and $\mathrm{EM}'_{\mathrm{KDC1}}$. $\mathrm{EM}_{\mathrm{KDC1}}$ is used exclusively for the session between KDC1 and $GW1$ while $\mathrm{EM}'_{\mathrm{KDC1}}$ is used solely for the session between KDC1 and KDC2.

$$\mathrm{KDC1} \rightarrow GW1 : \{ss, GW2, \mathrm{KDC2}, A, B, \mathrm{KDC1}, (EM'_{\mathrm{KDC1}})_{\mathrm{MK}_{C1}}, \mathrm{EM}_{\mathrm{KDC1}}\}.$$

The reason for two different event markers is the following. If the event marker used for the session between KDC1 and KDC2 is identical to the one used between KDC1 and $GW1$, this event marker will appear in both clear and in the ciphered form. Since these two pieces of information can be obtained by an intruder by wiretapping, a known plaintext attack can be mounted and security will be compromised.

Step 4) $GW1$ generates a new event marker $\mathrm{EM}_{\mathrm{GW1}}$ and

$$GW1 \rightarrow GW2 : \{ss, \mathrm{KDC2}, A, B, \mathrm{KDC1}, (\mathrm{EM}'_{\mathrm{KDC1}})_{\mathrm{MK}_{C1}}, \mathrm{EM}_{GW1}\}.$$

EM_{GW1} is the event marker used for the session between $GW1$ and $GW2$.

Step 5) $GW2$ generates an event marker EM_{GW2} which is used exclusively for the session with KDC2 and

$$GW2 \rightarrow \mathrm{KDC2} : \{ss, A, B, GW1, \mathrm{KDC1}, (\mathrm{EM}'_{\mathrm{KDC1}})_{\mathrm{MK}_{C1}}, \mathrm{EM}_{GW2}\}.$$

Step 6) Upon receiving the above message, KDC2 learns that KDC1 is attempting to establish a secure session between PCUA on his network and PCUB on the network controlled by KDC2. Now KDC2 checks the privilege status of PCUB to engage in encrypted sessions and if privileged, generates a session key SK and an event marker $\mathrm{EM}_{\mathrm{KDC2}}$. Further, he requests an exchange key from LHCC by transmitting the following message:

$$\mathrm{KDC2} \rightarrow \mathrm{LHCC} : \{(\mathrm{REK}, \mathrm{KDC1}, (\mathrm{EM}'_{\mathrm{KDC1}})_{\mathrm{MK}_{C1}}, \mathrm{EM}_{\mathrm{KDC2}})_{\mathrm{MK}_{C2}}\}$$

where REK denotes the request for exchange key.

Step 7) LHCC decrypts the received message under MK_{C2}, learns of the identity of KDC1 with whom KDC2 is desirous of having a secure exchange, generates the exchange key EK_{21} and

$$\mathrm{LHCC} \rightarrow \mathrm{KDC1} : \{(\mathrm{EK}_{21}, \mathrm{KDC2}, \mathrm{EM}'_{\mathrm{KDC1}})_{\mathrm{MK}_{C1}}\}$$

$$\mathrm{LHCC} \rightarrow \mathrm{KDC2} : \{(\mathrm{EK}_{21}, \mathrm{KDC1}, \mathrm{EM}_{\mathrm{KDC2}})_{\mathrm{MK}_{C2}}\}.$$

KDC1 and KDC2 decrypt the received messages and obtain EK_{21}.

Step 8) KDC2 → $GW2$: $\{GW1, \mathrm{KDC1}, \mathrm{A}, \mathrm{B}, \mathrm{KDC2}, (\mathrm{SK}, (\mathrm{EM}'_{\mathrm{KDC1}})_{\mathrm{MKC1}})\mathrm{EK}_{21}, \mathrm{EM}_{GW2}\}$.

Step 9) $GW2$ checks EM_{GW2} and

$$GW2 \rightarrow GW1 : \{\mathrm{KDC1}, A, B, \mathrm{KDC2}, (SK, (\mathrm{EM}'_{\mathrm{KDC1}})_{\mathrm{MK}_{C1}})_{\mathrm{EK}_{21}}, \mathrm{EM}_{GW1}\}.$$

Step 10) $GW1$ checks EM_{GW1} and

$$GW1 \rightarrow \mathrm{KDC1} : \{A, B, \mathrm{KDC2}, (SK, (\mathrm{EM}'_{\mathrm{KDC1}})_{\mathrm{MK}_{C1}})_{\mathrm{EK}_{21}}, \mathrm{EM}_{\mathrm{KDC1}}\}.$$

Upon receipt of this message, KDC1 decrypts $(\mathrm{SK}, (\mathrm{EM}'_{\mathrm{KDC1}})_{\mathrm{MK}_{\mathrm{C1}}})_{\mathrm{EK}_{21}}$ under EK_{21}, further decrypts $(\mathrm{EM}'_{\mathrm{KDC1}})_{\mathrm{MK}_{\mathrm{C1}}}$ under MK_{C1}, checks $\mathrm{EM}'_{\mathrm{KDC1}}$ and stores SK.

At the end of this step, both KDC1 and KDC2 have SK and are ready to distribute this key to the two end communicants PCUA and PCUB, which will be done in parallel on the two local networks using the network specific protocols as follows.

Step 11) a) KDC1 makes a remote secure call to $GW1$ for PCUA to transmit the following message:

$$A \rightarrow GW1 : \{A, (\mathrm{EM}_a)_{\mathrm{MK}_a}\}.$$

b) KDC2 makes a remote secure call to $GW2$ for PCUB to transmit the following message:

$$B \rightarrow GW2 : \{B, (\mathrm{EM}_b)_{\mathrm{MK}_b}\}.$$

Step 12)

a) $GW1 \rightarrow \mathrm{KDC1} : \{A, (\mathrm{EM}_a)_{\mathrm{MK}_a}, GW1, \mathrm{EM}_{GW1}\}$.

b) $GW2 \rightarrow \mathrm{KDC2} : \{B, (\mathrm{EM}_b)_{\mathrm{MK}_b}, GW2, \mathrm{EM}_{GW2}\}$.

It may be noted that the event markers EM_{GW1} and EM_{GW2} are the same ones that the two gateways used earlier in Step 4) and Step 5), respectively, since these steps are part of the same session establishment attempt.

Step 13)

a) $\mathrm{KDC1} \rightarrow GW1 : \{(\mathrm{SK}, GW1, \mathrm{EM}_a)_{\mathrm{MK}_a}, (A, \mathrm{EM}_{GW1})\}$.

b) $\mathrm{KDC2} \rightarrow GW2 : \{(\mathrm{SK}, GW2, \mathrm{EM}_b)_{\mathrm{MK}_b}, (B, \mathrm{EM}_{GW2})\}$.

Step 14) Upon receipt of the above messages, $GW1$ and $GW2$ retain the second part of the messages and forward the remainder to PCUA and PCUB, i.e.,

a) $GW1 \rightarrow A : \{(\text{SK}, GW1, \text{EM}_a)_{\text{MK}_a}\}$,

b) $GW2 \rightarrow B : \{(\text{SK}, GW2, \text{EM}_b)_{\text{MK}_b}\}$.

PCUA and PCUB decrypt the received messages under their master keys and obtain SK. The two PCU's are now ready to begin the secure session.

B. Some Discussions and Remarks

The detailed interactions described above are representative of the application of the present hierarchical approach to a specific internet scenario. These illustrate how the intranet protocols supported by the existing networks within the internet system can be utilized to build up a hierarchical key management scheme to support end-to-end encrypted sessions between source-destination pairs in the internet system. Several modifications of the specific exchanges in the various steps listed above can be made to realize additional resistance to penetration and other performance features. Due to limitations of space, these will not be given here; they may, however, be found in [15]. Mention will only be made of one illustrative modification in the following.

The transmission of EK_{21} by LHCC to KDC1 and KDC2 in Step 7) will be performed in parallel and might require a further handshaking between KDC1 and KDC2 to ensure that KDC1 is in receipt of EK_{21} before KDC2 transmits the message in Step 8). This may not be highly attractive due to the longer delays on the long-haul network. Alternately, Steps 7)–10) can be modified to have LHCC route EK_{21} to KDC1 via KDC2 as in the following:

Step 7')

$$\text{LHCC} \rightarrow \text{KDC2} : \{(\text{EK}_{21}, \text{KDC1}, \text{EM}_{\text{KDC2}})_{\text{MK}_{C2}}, (\text{EK}_{21}, \text{KDC2}, \text{EM}'_{\text{KDC1}})_{\text{MK}_{C1}}\}.$$

Step 8')

$$\text{KDC2} \rightarrow GW2 : \{GW1, \text{KDC1}, A, B, \text{KDC2}, (\text{SK})_{\text{EK}_{21}}, (\text{EK}_{21}, \text{KDC2}, \text{EM}'_{\text{KDC1}})_{\text{MK}_{C1}}, \text{EM}_{GW2}\}.$$

Step 9')

$$GW2 \rightarrow GW1 : \{\text{KDC1}, A, B, \text{KDC2}, (\text{SK})_{\text{EK}_{21}}, (\text{EK}_{21}, \text{KDC2}, \text{EM}'_{\text{KDC1}})_{\text{MK}_{C1}}, \text{EM}_{GW1}\}.$$

Step 10')

$$GW1 \rightarrow \text{KDC1} : \{A, B, \text{KDC2}, (\text{SK})_{\text{EK}_{21}}, (\text{EK}_{21}, \text{KDC2}, \text{EM}'_{\text{KDC1}})_{\text{MK}_{C1}}, \text{EM}_{\text{KDC1}}\}.$$

IV. Verification of Security of the Present Scheme

The important problem of verifying the security in a communication network employing a specific key management scheme has been dealt with in the literature by several different methods. A majority of papers that concern the design of new schemes for key generation and distribution adopt the approach of advancing verbal arguments and attempting to construct scenarios where a specific scheme could be proved insecure from which improvements to eliminate the identified problems could be developed. An example of this approach is the identification of certain improvements to the classic key management protocol of Needham and Schroeder [3], by Denning and Sacco [4], and later by Bauer *et al.*, [5] by constructing specific scenarios where the original protocol can be shown to be not secure. Such an approach, while simple to comprehend, requires a careful and painstaking identification of all scenarios where security can be breached. This is not a shortcoming of the approach, however, since the demonstration of the security of a system often reduces to proving that the system is not insecure under certain stated conditions.

In the very recent times, there are a few attempts at the formal verification of security in distributed systems by methods that employ more mathematical and systematic procedures. One of these verification methodologies, given by Britton [17], uses a representation of the system as a set of concurrent processes each modeled by a finite state machine and employs the commercially available VERSUS verification software package [18] for checking the correctness of assertions and functional specifications. A more mathematical approach, given by Kasami *et al.* [19], employs arguments from combinatory logic to prove the security of a system defined by a finite number of axiomatic operations.

In this section, we shall demonstrate the security of a communication network (internet system) where the present hierarchical key management scheme is used for end-to-end encryption, by following an approach inspired by the work of Kasami *et al.* [19]. We will begin by introducing some notations which will help in precisely defining the various operations underlying the scheme for key management. These operations will then be described by a set of axioms. We will then define a class of information sets that could be available to a potential intruder without compromising the security of the system in the sense that the intruder will not have access to the message being protected, the session key SK and the exchange keys used for transmitting SK by transforming the intercepted messages using the properties induced from the axioms which define the operations governing the system.[3]

A. Axiomatic Representation of the Operations

Let $\mathcal{N}$ denote the set of all communication nodes included in the internet system, $\mathcal{A}$ denote the set of authentication servers and $\mathcal{G}$ denote the set of gateways. Let n_p denote the identifier (address or logical name) of a node $P \in \mathcal{N}$.

Let $E(K, x)$ denote the function of encrypting a message x with a key K and $D(K, x)$ denote the function of decrypting message x with key K. Let MK_p denote the master key of node $P \in \mathcal{N}$, HK_i denote the storage key used for storing the master key table by the authentication server $\text{AS}_i \in \mathcal{A}$, EK_{ij} denote the inter-AS exchange key used by AS_i and $\text{AS}_j \in \mathcal{A}$ and SK denote the session key for any specific session under consideration.

As described earlier, SK is generated by a random number generator inside an authentication server as a function of its current internal state. Let $\mathcal{S}_j$ denote the set of the internal states of the random number generator inside $\text{AS}_j \in \mathcal{A}$ and let ST(s) define the state transition function from state $s \in \mathcal{S}_j$. Also, let $OP(s)$ define the output function of the random number generator for a state $s \in \mathcal{S}_j$. Note that ST: $\mathcal{S}_j \rightarrow \mathcal{S}_j$ and OP: $\mathcal{S}_j \rightarrow \mathcal{K}$ where $\mathcal{K}$ is the set of all session keys.

In the execution of the basic internet protocol outlined in Section II-C, several operations are performed at the individual steps. For purposes of concisely describing these in our

[3] It should be noted that the notion of security considered here is in this limited sense, since our present interest is only to demonstrate this property for the illustrative intranetwork protocols selected in the design process. Other stronger security requirements (such as resistance to active attacks) are not intended to be considered here as they will require using a corresponding intranetwork protocol (available in the literature) that foils these attacks within the present hierarchical structure and using a correspondingly expanded set of axioms to include the additional functions. One may, however, note that the encryption and the passing of the event markers provides authentication in the sense that this is a verifiable part of the message, thus preventing some forms of active attacks.

later work, we shall use a functional representation of each of these operations as in the following.

For increased security, we shall assume that a session key received at any node $P \in \mathfrak{N}$ will exist in the appropriate register only in an encrypted form, encrypted under its master key MK_p. This is to prevent the compromise of SK due to an intruder breaking into the register. In such a case, when the actual message transmission begins, the encryption of a message m under SK at node P must be preceded by the decryption under MK_p of the stored SK. These operations will be described in an input-output functional form as

$$\mathfrak{F}_{\mathrm{ED}}(E\,(\mathrm{MK}_p,\ \mathrm{SK}),\ m) = E\,(\mathrm{SK},\ m). \tag{22}$$

Note that the term on the left-hand side is equal to $E\,(D(\mathrm{MK}_p,\ E(\mathrm{MK}_p,\ \mathrm{SK})),\ m)$ which results in $E(\mathrm{SK},\ m)$. For a decryption of the received enciphered message $E(\mathrm{SK},\ m)$, almost identical operations are to be performed and these will be denoted as

$$\mathfrak{F}_{\mathrm{DD}}(E\,(\mathrm{MK}_p,\ \mathrm{SK}),\ E\,(\mathrm{SK},\ m)) = m. \tag{23}$$

Thus $\mathfrak{F}_{\mathrm{ED}}$ and $\mathfrak{F}_{\mathrm{DD}}$ describe the operations of *encryption following a decryption* and *decryption following another decryption*, respectively.

For the generation of a session key SK at an authentication server $\mathrm{AS}_j \in \mathfrak{A}$ when a message of the form $n_a \cdot n_b \cdot \mathrm{EM}_{asi}$ is received[4] from AS_i as in Step 2),[5] and to send the response $E(\mathrm{EK}_{ij},\ \mathrm{SK} \cdot n_a \cdot \mathrm{EM}_{asi})$ to AS_i as in Step 3), the following operations are to be performed: 1) SK is generated using the current internal state s_j of the random number generator at AS_j (i.e., $\mathrm{SK} = OP(s_j)$), with the resulting new value of the state being $\mathrm{ST}(s_j)$. 2) Next, n_a and EM_{asi} are extracted from the concatenated message $n_a \cdot n_b \cdot \mathrm{EM}_{asi}$ using the extraction operations

$$\mathfrak{F}^{\,l_1}_{\mathrm{SF}}(n_a\ \cdot\ n_b\ \cdot\ \mathrm{EM}_{asi}) = n_a \tag{24}$$

$$\mathfrak{F}^{\,l_1}_{\mathrm{SL}}(n_a\ \cdot\ n_b\ \cdot\ \mathrm{EM}_{asi}) = \mathrm{EM}_{asi} \tag{25}$$

which define the *extraction* of the first l_1 bits and the last l_1 bits, respectively, of the input message. We shall assume that all event markers and node identifiers are represented in a field of length l_1. 3) After this step, a concatenation of messages is performed to obtain $\mathrm{SK} \cdot n_a \cdot \mathrm{EM}_{asi}$ which is to be encrypted under EK_{ij} for transmission to AS_i.If we assume that EK_{ij} is stored at AS_j encrypted under the storage key HK_j (i.e., $E(\mathrm{HK}_j,\ \mathrm{EK}_{ij})$ will be available), a decryption operation must be performed as a preceding step. (4)AS_j after transmitting this message holds a copy of SK encrypted under HK_j (i.e., $E(\mathrm{HK}_j,\ \mathrm{SK})$ is stored). We will combine all of the operations described above in (1)–(4) by a functional representation

$$\mathfrak{F}_{\mathrm{GT}}(E\,(\mathrm{HK}_j,\ \mathrm{EK}_{ij}),\ n_a\ \cdot\ n_b\ \cdot\ \mathrm{EM}_{asi},\ s_j)$$

$$= \{E\,(\mathrm{EK}_{ij},\ \mathrm{SK}\ \cdot\ n_a\ \cdot\ \mathrm{EM}_{asi}),\ E\,(\mathrm{HK}_j,\ \mathrm{SK}),\ \mathrm{ST}(s_j)\}. \tag{26}$$

Thus, $\mathfrak{F}_{\mathrm{GT}}$ describes the operations of *generation and transmission* of SK.

When AS_i receives $E(\mathrm{EK}_{ij},\ \mathrm{SK} \cdot n_a \cdot \mathrm{EM}_{asi})$, it will perform the following operations: 1) From the stored $E(\mathrm{HK}_i,\ \mathrm{EK}_{ij})$ in its memory, a decryption under HK_i to obtain EK_{ij} is performed. 2) The received message is decrypted using EK_{ij} to obtain $\mathrm{SK} \cdot n_a \cdot \mathrm{EM}_{asi}$. 3) Next, SK is extracted using the extraction operation

$$\mathfrak{F}^{\,l_2}_{\mathrm{SF}}(\mathrm{SK}\ \cdot\ n_a\ \cdot\ \mathrm{EM}_{asi}) = \mathrm{SK} \tag{27}$$

which defines the extraction of the first l_2 bits of the input message. We shall assume that all session keys are represented in a field of length l_2. 4) The obtained SK is stored being encrypted under HK_i (i.e., $E(\mathrm{HK}_i,\ \mathrm{SK})$ is stored). The operations described in (1)–(4) above can be functionally represented by

$$\mathfrak{F}_{\mathrm{RSA}}(E\,(\mathrm{HK}_i,\ \mathrm{EK}_{ij}),\ E\,(\mathrm{EK}_{ij},\ \mathrm{SK}\ \cdot\ n_a\ \cdot\ \mathrm{EM}_{asi}))$$

$$= E\,(\mathrm{HK}_i,\ \mathrm{SK}). \tag{28}$$

Thus, $\mathfrak{F}_{\mathrm{RSA}}$ describes the operation of *reenciphering SK for storage at an authentication server*. After SK is obtained at AS_i, the following two steps, viz. Step 4) and Step 5), of the basic protocol do not involve the transmission of secure messages and hence no special operations are required to describe these.

In Step 6), each authentication server is required to transmit certain messages to the gateway GW. Specifically, AS_i, after receiving from GW the messages n_a, EM_a, GW, and EM_{gw} performs the following operations: 1) It fetches $E(\mathrm{HK}_i,\ \mathrm{SK})$ and $E(\mathrm{HK}_i,\ \mathrm{MK}_a)$ from memory and obtains SK and MK_a. 2) Then, it concatenates messages to form $\mathrm{SK} \cdot GW \cdot \mathrm{EM}_a$ and $n_a \cdot \mathrm{EM}_{gw}$. 3) Now, it encrypts $\mathrm{SK} \cdot \mathrm{GW} \cdot \mathrm{EM}_a$ under MK_a to obtain $E(\mathrm{MK}_a,\ \mathrm{SK} \cdot GW \cdot \mathrm{EM}_a)$ and transmits this and $n_a \cdot \mathrm{EM}_{gw}$ to GW. The above set of operations can be functionally represented by

$$\mathfrak{F}_{\mathrm{RT}}(E\,(\mathrm{HK}_i,\ \mathrm{SK}),\ E\,(\mathrm{HK}_i,\ \mathrm{MK}_a),\ n_a,\ \mathrm{EM}_a,\ GW,\ \mathrm{EM}_{gw})$$

$$= \{E\,(\mathrm{MK}_a,\ \mathrm{SK}\ \cdot\ GW\ \cdot\ \mathrm{EM}_a),\ n_a\ \cdot\ \mathrm{EM}_{gw}\}. \tag{29}$$

Thus, $\mathfrak{F}_{\mathrm{RT}}$ describes the operation of *reenciphering SK for transmission by an authentication server*. The transmission from AS_j to GW is identical to the above and is described by

$$\mathfrak{F}_{\mathrm{RT}}(E\,(\mathrm{HK}_j,\ \mathrm{SK}),\ E\,(\mathrm{HK}_j,\ \mathrm{MK}_b),\ n_b,\ \mathrm{EM}_b,\ GW,\ \mathrm{EM}_{gw})$$

$$= \{E\,(\mathrm{MK}_b,\ \mathrm{SK}\ \cdot\ GW\ \cdot\ \mathrm{EM}_b),\ n_b\ \cdot\ \mathrm{EM}_{gw}\}. \tag{30}$$

Finally, in Step 7), when node A receives $E(\mathrm{MK}_a,\ \mathrm{SK} \cdot GW \cdot \mathrm{EM}_a)$ from GW, it will perform the following sequence of operations: 1) The received message is decrypted using MK_a to obtain $\mathrm{SK} \cdot GW \cdot \mathrm{EM}_a$. 2) Then, SK is extracted using the extraction operation

$$\mathfrak{F}^{\,l_2}_{\mathrm{SF}}(\mathrm{SK}\ \cdot\ GW\ \cdot\ \mathrm{EM}_a) = \mathrm{SK}$$

as described in (27). 3) The obtained SK is encrypted under MK_a for storage, i.e., $E(\mathrm{MK}_a,\ \mathrm{SK})$ is stored). The above sequence of operations can be functionally represented by

$$\mathfrak{F}_{\mathrm{RSN}}(\mathrm{MK}_a,\ E\,(\mathrm{MK}_a,\ \mathrm{SK}\ \cdot\ GW\ \cdot\ \mathrm{EM}_a)) = E\,(\mathrm{MK}_a,\ \mathrm{SK}). \tag{31}$$

Thus, $\mathfrak{F}_{\mathrm{RSN}}$ describes the operation of *reenciphering SK for storage at a node*.

In this step, an identical operation is performed in parallel at node B, which is described by

$$\mathfrak{F}_{\mathrm{RSN}}(\mathrm{MK}_b,\ E\,(\mathrm{MK}_b,\ \mathrm{SK}\ \cdot\ GW\ \cdot\ \mathrm{EM}_b)) = E\,(\mathrm{MK}_b,\ \mathrm{SK}). \tag{32}$$

To aid in the verification of security to be conducted in the next section, we shall summarize all of the above described operations in a slightly more general format as a set of axioms given by $A1$–$A10$ below.

$A1$: $E(x,\ D(x,\ y))\ = =\ y$.
$A2$: $D(x,\ E(x,\ y))\ = =\ y$.
$A3$: $\mathfrak{F}_{\mathrm{ED}}(x,\ y)\ = =\ E(D(\mathrm{MK}_p,\ x),\ y)$.
$A4$: $\mathfrak{F}_{\mathrm{DD}}(x,\ y)\ = =\ D(D(\mathrm{MK}_p,\ x),\ y)$.
$A5$: $\mathfrak{F}^{l}_{\mathrm{SF}}(x_1 \cdot x_2 \cdot x_3)\ = =\ x_1$.

[4] Note that n_a and n_b are the identifiers of nodes A and B and $n_a \cdot n_b \cdot \mathrm{EM}_{asi}$ denotes the concatenated message corresponding to $\{A,\ B,\ \mathrm{EM}_{asi}\}$ received at AS_j.

[5] The various steps recalled here and in the following refer to those of the basic internet protocol outlined in Section II-C.

$A6$: $\mathfrak{F}^1_{\mathrm{SL}}(x_1 \cdot x_2 \cdot x_3) == x_3$.

$A7$: $\mathfrak{F}_{\mathrm{GT}}(x, y, z) == \{E(D(\mathrm{HK}_u, x), \mathrm{OP}(s) \cdot \mathfrak{F}^1_{\mathrm{SF}}(y) \cdot \mathfrak{F}^1_{\mathrm{SL}}(y)), E(\mathrm{HK}_u, OP(s)), \mathrm{ST}(s)\}$.

$A8$: $\mathfrak{F}_{\mathrm{RSA}}(x, y) == E(\mathrm{HK}_u, \mathfrak{F}^1_{\mathrm{SF}}(D(D(\mathrm{HK}_u, x), y)))$.

$A9$: $\mathfrak{F}_{\mathrm{RT}}(x, y, z_1, z_2, z_3, z_4) == \{E(D\mathrm{HK}_u, y), D(\mathrm{HK}_u, x) \cdot z_3 \cdot z_2), z_1 \cdot z_4\}$.

$A10$: $\mathfrak{F}_{\mathrm{RSN}}(x, y) == E(x, \mathfrak{F}^1_{\mathrm{SF}}(D(x, y)))$.

It may be noted that $A1$ and $A2$ describe the basic encryption and decryption algorithms, while the rest of the axioms describe the operations defined in (22)–(32) above. The variables used in the statements of the axioms (x, y, x_1, x_2, etc.) are generic variables and HK_u represents the storage key of a general authentication server $\mathrm{AS}_u \in \mathcal{A}$. Also, note that for $A5$ and $A6$ to be valid, x_1 and x_3 are assumed to be representable in a field of length l [l can take values l_1 and l_2 as in (24), (25), and (27)].

B. Security Verification

In the verification of the security of the key management scheme, we will focus on a specific transaction between two end communicants, node A and B lying on the networks LN_i and LN_j, respectively, as described in the scenario outlined in Section II-A. The establishment of an end-to-end secure session for this transaction requires that the session key SK be available at node A and node B (the principals) and at the authentication servers AS_i and AS_j of the two networks (the aides). Outside of this group of four units, SK should not be available to any other unit in the internet system, since any other node (on LN_i or LN_j or any other network), any gateway or any other authentication server in the internet system could be used by a potential intruder for breaching the security of the session and consequently, we will regard these units as potential adversaries for the specific session under consideration. Since these units will also be involved in other sessions which could be of a secure nature, it is necessary that the encryption/decryption algorithms being used and the key management operations described by the axioms be available to these units. In such an environment, the demonstration of the security of the system reduces to the following question: if the "privileged information" (plaintext being transacted, session key SK, master keys, exchange keys, etc.) specific to the session under consideration is restricted only to the principals and the aides, can an adversary using data other than this privileged information derive this from a finite number of operations (induced by the axioms) available to him? If this question is not answered in the affirmative, we will say that the key management scheme is secure in the sense of protecting the privileged information from disclosure.

In this section, we shall begin by defining the information sets that can be obtained by a potential adversary, which could be regarded "nonprivileged" for the session under consideration. We will then demonstrate that this information cannot be reduced by using the available operations to privileged information.

For precisely describing the privileged and the nonprivileged information sets, let a *term* of an information set denote a variable in an axiom or a function defining any operation.

The privileged information set $\mathcal{I}_P$ contains the following terms: 1) the plaintext message m, 2) the session key SK, 3) the master keys MK_a and MK_b, 4) the storage keys HK_i and HK_j of AS_i and AS_j, 5) exchange key EK_{ij} and 6) the internal states of the random number generators at AS_i and AS_j. Note that a knowledge of any of the above terms enables m or SK to be known.

The nonprivileged information set $\mathcal{I}_{NP}$ can be defined to include the following terms:

*I*1) n_p, $\mathrm{EM}_p \; \forall \; P \in \mathcal{N}$; $\mathrm{EM}_u \; \forall \; u \ni \mathrm{AS}_u \in \mathcal{A}$; $\mathrm{EM}_{gw} \; \forall \; gw \in \mathcal{G}$.

*I*2) $\mathrm{MK}_p \; \forall \; P \in \mathcal{N} - \{A, B\}$; $\mathrm{HK}_u \; \forall \; u \ni \mathrm{AS}_u \in \mathcal{A} - \{\mathrm{AS}_i, \mathrm{AS}_j\}$; $\mathrm{EK}_{uv} \; \forall \; u, v (u \neq v) \ni \mathrm{AS}_u, \mathrm{AS}_v \in \mathcal{A} - \{\mathrm{AS}_i, \mathrm{AS}_j\}$.

*I*3) $s_u \; \forall \; u \ni \mathrm{AS}_u \in \mathcal{A} - \{\mathrm{AS}_i, \mathrm{AS}_j\}$.

*I*4) $E(\mathrm{HK}_u, \mathrm{EK}_{uv}) \; \forall \; u, v (u \neq v) \ni \mathrm{AS}_u, \mathrm{AS}_v \in \mathcal{A}$; $E(\mathrm{HK}_u, \mathrm{MK}_p) \; \forall \; u \ni \mathrm{AS}_u \in \mathcal{A}$ and $\forall \; P \in \mathcal{N}$.

Note that this is the information stored in the memory of the authentication servers.

*I*5) $E(\mathrm{SK}, m)$ which is the encrypted message that can be tapped during flight on the internet system.

*I*6) $E(\mathrm{EK}_{ij}, OP(s) \cdot n_a \cdot \mathrm{EM}_{asi})$, $E(\mathrm{HK}_j, OP(s)) \; \forall s \in \mathcal{S}_q$, $q = i, j$.

Note that these are the first two outputs available externally from the $\mathfrak{F}_{\mathrm{GT}}$ operation (The third output, $\mathrm{ST}(s)$, which is internal to the random number generator is not available).

*I*7) $E(\mathrm{MK}_p, \mathrm{SK} \cdot GW \cdot \mathrm{EM}_p)$, $n_p \cdot \mathrm{EM}_{gw}$ for $p = a, b$. Note that these two terms are the outputs of the $\mathfrak{F}_{\mathrm{RT}}$ operation.

*I*8) $E(\mathrm{HK}_u, \mathrm{SK})$ for $u = i, j$, which is the output of $\mathfrak{F}_{\mathrm{RSA}}$ operation.

*I*9) $E(\mathrm{MK}_p, \mathrm{SK})$ for $p = a, b$, which define the output of $\mathfrak{F}_{\mathrm{RSN}}$ operations at A and B.

*I*10) All functions described in axioms $A1$ through $A10$.

Starting with the information available to him, the adversary will attempt to transform this by repeated applications of the available functions (i.e., functions described in axioms $A1$ to $A10$) in order to expand his information set. Let $\mathcal{I}^r_{\mathrm{NP}}$ denote the expanded information set obtained after r applications of these functions on the terms in $\mathcal{I}_{\mathrm{NP}}$. Clearly, $\mathcal{I}_{\mathrm{NP}} \subseteq \mathcal{I}^1_{\mathrm{NP}} \subseteq \mathcal{I}^2_{\mathrm{NP}} \cdots \subseteq \mathcal{I}^r_{\mathrm{NP}}$ and $\mathcal{I}^r_{\mathrm{NP}} - \mathcal{I}^{r-1}_{\mathrm{NP}}$ will be the set of new terms obtained at the rth application.

Let α be a positive integer such that $\mathcal{I}^{\alpha}_{\mathrm{NP}} - \mathcal{I}^{\alpha-1}_{\mathrm{NP}} = \phi$ and let $\mathfrak{J} = \mathcal{I}^{\alpha}_{\mathrm{NP}} \cup \mathcal{I}_P$. For a term $t \in \mathfrak{J}$, a transformation into a term $t' \in \mathfrak{J}$ by using any of the available functions is denoted $t \rightarrow t'$ and is called a *reduction of t into t'*. A sequence of reductions starting from a term $t = t_1 \in \mathfrak{J}$ of the form $t_1 \rightarrow t_2 \rightarrow t_3 \rightarrow \cdots$ is said to terminate at t_δ when $t_\delta = t_{\delta-1}$ for any $i = 1, 2, \cdots, (\delta - 1)$ and this sequence is denoted by $t \overset{*}{\rightarrow} t_\delta$.

With the above definitions, it is now possible to more formally define the security properties of the system resulting from the implementation of a key management scheme.

Definition: For any t such that $t \in \mathcal{I}^r_{\mathrm{NP}}$, $r = 0, 1, 2, \cdots, \alpha$ and $t \overset{*}{\rightarrow} t_\delta$ for $\delta > 0$, if α is finite and $t_{\delta-i} \notin \mathcal{I}_p \; \forall \; i = 0, 1, 2, \cdots \delta$ where $t_0 = t$, then the system is said to be secure in the sense of offering protection of privileged information from disclosure.

The demonstration of security of the present key management scheme can now be made using certain concepts from combinatory logic and mathematical induction principles. Specifically, since $\mathcal{I}_{\mathrm{NP}} \cap \mathcal{I}_P = \phi$, it suffices to show that a) α is finite and b) for any $t \in \mathcal{I}^r_{\mathrm{NP}} - \mathcal{I}^{r-1}_{\mathrm{NP}}$ for any r, $t \overset{*}{\rightarrow} t_\delta$ and $t_{\delta-i} \notin \mathcal{I}_P \; \forall i = 0, 1, 2, \cdots \delta$. The first condition can be simply established observing that the system described by the axiom set $A1$–$A10$ satisfies the Church-Rosser [6] property [20], [21]. For showing the second condition, one can start with each term in the set $\mathcal{I}^r_{\mathrm{NP}} - \mathcal{I}^{r-1}_{\mathrm{NP}}$ and generate all possible reductions from using the available functions to demonstrate that no term obtained from any reduction belongs to $\mathcal{I}_P$. For the sake of brevity, the details of these reductions are given only for an illustrative case of a specific term in $\mathcal{I}^r_{\mathrm{NP}} - \mathcal{I}^{r-1}_{\mathrm{NP}}$. The arguments are almost identical for the other terms; details of these are omitted here and can be found in [22].

Consider a term of the form $y^{(r)} = E(D(\mathrm{MK}_p, x^{(r-1)}), y^{(r-1)})$ which is an element of $\mathcal{I}^r_{\mathrm{NP}} - \mathcal{I}^{r-1}_{\mathrm{NP}}$. Note that $y^{(r)}$ is the resultant of the enciphering operation described by the axiom $A3$. For this operation to be possible, it is evident that $y^{(r-1)}$,

[6] The Church-Rosser theorem, initially given for lambda calculus, has been extensively used in the analysis of term rewriting systems. Very simply, a reduction is said to satisfy the Church-Rosser property if starting with any object, a unique irreducible object is reached.

$x^{(r-1)} \in \mathcal{G}_{\mathrm{NP}}^{r-1}$. From axiom $A1$ it now follows that $y^{(r-1)}$ must be of the form, $y^{(r-1)} = D(z, y^{(r-2)}$ where $y^{(r-2)} \in \mathcal{G}_{\mathrm{NP}}^{(r-2)}$. Hence, by induction one can show that $y^{(r-2)} \overset{*}{\rightarrow} y^{(0)} \in \mathcal{G}_{\mathrm{NP}}^{(0)} = \mathcal{G}_{\mathrm{NP}}$ and since $\mathcal{G}_P \cap \mathcal{G}_{\mathrm{NP}} = \phi$, $y^{(0)} \notin \mathcal{G}_P$.

The method of verification given above is quite general and can be used to verify the security of any other key management scheme. In particular, it can be employed to the modified key management protocols described in Section II-C and D for general internet systems. The detailed verification would however require an expanded set of axioms (to include the additional functions); nevertheless the basic principles would remain identical to those employed presently for verifying the security of the basic internet protocol.

V. Conclusions

The major contributions of this paper are the development of a new scheme for key management in internet environments to support end-to-end encryption and a method for the formal verification of the security of the resulting system. The key management scheme is of a hierarchical nature and is built upon the existing key distribution protocols (such as the ANSI X9.17) in the individual networks which form the lower levels of the hierarchy. At the higher level are the protocols between authentication servers and/or control centers for different regions dividing the internet system. Major strong features of this approach are that the scheme is simple to implement and does not require extensive modifications to the network specific functions (hardware and software) when a secure network is to be brought into the internet system. Several details on the implementation of the key management scheme have been given by considering a specific scenario of an internet system comprising of two remote Sytek LocalNet 20 networks interconnected via a Long-Haul Network (such as the DDN). A systematic method that uses an axiomatic approach is offered for verifying the security provided by the present scheme. Although the method is given here in a form tailored to demonstrate the security of an internet system implementing the basic internet protocol described in this paper, it is of a general form and could be used for the security verification of other key management schemes existing in the literature.

References

[1] W. F. Ehrsam, S. M. Matyas, C. H. Meyer, and W. L. Tuchman, "A cryptographic key management scheme for implementing the data encryption standard," *IBM Syst. J.*, vol. 17, pp. 106-125, Feb. 1978.

[2] S. M. Matyas and C. H. Meyer, "Generation, distribution, and installation of cryptographic keys," *IBM Syst. J.*, vol. 17, pp. 126-137, Feb. 1978.

[3] R. M. Needham and M. D. Schroeder, "Using encryption for authentication in large networks for computers" *Commun. ACM*, vol. 21, pp. 993-999, Dec., 1978.

[4] D. E. Denning and G. M. Sacco, "Timestamps in key distribution protocols," *Commun. ACM,* vol. 24, pp. 533-536, Aug. 1981.

[5] R. C. Bauer, T. A. Berson, and R. J. Feiertag, "A key distribution protocol using event markers," *ACM Trans. Comput. Syst.*, vol. 1, pp. 249-255, May 1983.

[6] LocalNet 50/100 Network Control Center—Key Distribution Center Option, Sytek, Inc., Preliminary Ref. Man. R1020.1/1-02A, Nov. 1983.

[7] V. Cerf and P. Kirstein, "Issues in packet network interconnection," *Proc. IEEE,* vol. 66, pp. 1386-1408, Nov. 1978.

[8] R. Martinez and M. K. Sundareshan, "Gateway design methodology for LocalNet 20-to-DDN interconnection," Rep. #2, Eng. Experiment Station, Univ. Arizona, Jan. 1984.

[9] H. Zimmerman, "OSI reference model—The ISO model of architecture for open system interconnection," *IEEE Trans. Commun.*, vol. COM-28, pp. 425-432, Apr. 1980.

[10] M. K. Sundareshan and K. H. Muralidhar, "Addressing in internet environments," Rep. #3, Eng. Experiment Station, Univ. Arizona, Nov. 1984.

[11] W. Diffie and M. E. Hellman, "New direction in cryptography," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 644-654, Nov. 1976.

[12] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public key crypto-systems," *Commun. ACM*, vol. 21, pp. 120-126, Feb. 1978.

[13] LocalNet 50/100 Network Control Center Ref. Man., Sytek, Inc., 1983.

[14] K. H. Muralidhar and M. K. Sundareshan, "On the decomposition of large communication networks for hierarchical control implementation," *IEEE Trans. Commun.*, vol. COM-34, pp. 985-988, Oct. 1986.

[15] M. K. Sundareshan and W. P. Lu, "Encryption key management for secure communication in internet environments," Rep. DAEA 18-85-K-0065, Eng. Experiment Station, Univ. Arizona, Jan. 1986.

[16] ANSI X 9.17 Financial Institution Key Management (Wholesale) Standard, American Banker Association, Washington, DC, Apr. 1985.

[17] D. E. Britton, "Formal verification of a secure network with end-to-end encryption," in *Proc. 1984 IEEE Symp. Security Privacy*, pp. 154-166.

[18] B. Marick, "The VERUS design verification system," in *Proc. 1983 Symp. Security Privacy*, pp. 150-157.

[19] T. Kasami, S. Yamamura, and K. Mori, "A key management scheme for end-to-end encryption and a formal verification of its security," *Syst., Comput., Cont.*, vol. 13, pp. 59-69, May-June, 1982.

[20] G. Huet, "Confluent reductions: Abstract properties and applications to term rewriting systems," *J. ACM*, vol. 27, pp. 797-821, Oct., 1980.

[21] R. Sethi, "Testing for the Church-Rosser property," *J. ACM*, vol. 21, pp. 671-679, Oct. 1974.

[22] W. P. Lu, "Security of communication in computer networks," Ph.D. dissertation, Univ. Arizona, Tucson, Aug. 1986.

★

Wen-Pai Lu (S'78-M'86) was born in Taiwan in 1957. He received the B.S. and M.E. degrees, both in electrical engineering, from the University of Tennessee, Knoxville, in 1978 and 1981, respectively, and the Ph.D. degree in electrical engineering from the University of Arizona, Tucson, in 1986.

From 1982 to 1983, he was a Systems Engineer at the Fairchild Semiconductor (S) Pte. Ltd., Singapore. Between 1983 and 1986, he worked as a Research Assistant while he was pursuing the Ph.D. degree, and was engaged in the research of secure communication in internetworking environments for army networks, and the research of implementing the security of communications in public telecommunication networks. In 1986, he joined the AT&T Bell Laboratories, Holmdel, NJ, where he worked on planning new services for AT&T networks. He was also responsible for implementing several prototype systems for future network services. Presently, he is developing a ptototype system for the ACD service. His current interests include computer system security, network security, access control, database security, communication network design and protocol, and ISDN.

Dr. Lu is currently a member of IEEE 802.10 Standard for Interoperable LAN Security (SILS) working group.

★

Malur K. Sundareshan (M'77) received the B.E. degree in electrical engineering from Bangalore University, Bangalore, India, in 1966, and the M.E. and Ph.D. degrees in electrical engineering from the Indian Institute of Science, Bangalore, India, in 1969 and 1973, respectively.

Between 1973 and 1976, he held various visiting faculty positions at the Indian Institute of Science, Bangalore, at the University of Santa Clara, CA, and at Concordia University, Montreal, PQ, Canada. From 1976-1981, he was on the faculty of the Department of Electrical Engineering, University of Minnesota, Minneapolis. Since 1981, he has been on the faculty of the Department of Electrical and Computer Engineering, University of Arizona, Tucson, where he is a Professor and the Chairman of the Systems Group comprising of communications, control systems, and signal processing faculty. His current research interests are in large-scale systems, communication networks, adaptive control and estimation, and statistical signal processing.