Reprinted from

COMPUTER NETWORKS and ISDN SYSTEMS

Computer Networks and ISDN Systems 26 (1994) 1523–1531

A proposed mechanism for implementation of non-discretionary access controls in a network environment

Rayford B. Vaughn [a], Hossein Saiedian [b,*], Elizabeth A. Unger [c]

[a] *U.S. Army Info. Systems Center, Fort Belvoir, Virginia 22060, USA*
[b] *Department of Computer Science, University of Nebraska, Omaha, Nebraska 68182, USA*
[c] *Computing and Info. Sciences, Kansas State University, Manhattan, Kansas 66506, USA*

Received 3 March 1992; revised version received 31 March 1993

ELSEVIER

ELSEVIER

Computer Networks and ISDN Systems 26 (1994) 1523–1531

COMPUTER NETWORKS and ISDN SYSTEMS

A proposed mechanism for implementation of non-discretionary access controls in a network environment

Rayford B. Vaughn [a], Hossein Saiedian [b,*], Elizabeth A. Unger [c]

[a] *U.S. Army Info. Systems Center, Fort Belvoir, Virginia 22060, USA*
[b] *Department of Computer Science, University of Nebraska, Omaha, Nebraska 68182, USA*
[c] *Computing and Info. Sciences, Kansas State University, Manhattan, Kansas 66506, USA*

Received 3 March 1992; revised version received 31 March 1993

Abstract

This paper investigates moving Lampson's reference monitor abstraction from the single system environment to a range of networked distributed systems which include interconnected office information systems. It suggests modifying our implementation of the abstraction from the traditional security kernel to a dual approach using a basic, node level reference monitor and a system level reference monitor that we choose to call a sentinel. An argument is presented that the sentinel meets the requirements of a reference monitor in that it provides separation, mediation, and can be formally verified. The approach to installing a sentinel is viewed as top down with great emphasis on the security mode implemented at each participating node.

Key words: Non-discretionary access controls; Office systems; Reference monitor; Distributed systems; Security mechanisms, Sentinel

1. Introduction

With the increasing applications of computer networks, security of information transported through networks (as well as security of network resources) have become the focus of concern for both network users and managers. As the result, computer network security has been the subject of a great deal of attention in recent years [5–7,11,12,16]. A recent issue of *Computer Networks and ISDN Systems* (**22**(5)), was devoted to the network security and included a number of excellent articles, including Refs. [7] and [11].

This paper is also about network security and addresses the enforcement of non-discretionary security control mechanism in a network environment. The work reported in this paper has emerged from research work in the area of Office Information Systems (OIS) security [14,15]. We explore the possibility of extending Lampson's well-defined and widely used reference monitor concept [8] from a single system domain to the distributed system environment. The background of this issue, the theory behind the reference monitor abstraction and its implementation, and the special considerations one faces within a dis-

* Corresponding author.

SSDI 0169-7552(94)E0094-U

tributed systems environment will be examined. In discussing the extension of the reference monitor abstraction into distributed systems, certain assumptions need to be made to avoid confusing the problem with less germane issues. Our assumptions are listed below:

- First, we assume adequate, acceptable physical protection of each of the distributed system components and their attached peripheral devices. This is considered a reasonable assumption and one that would command a high level of attention in a practical implementation.
- Second, we assume the use of adequate and appropriate cryptography which will insure the absolute secrecy and privacy of the information transmitted from the physical protection boundary of the sending component to the protection boundary of the receiving component. We are not concerned with the method of information relay or the type of cipher mechanism used. This issue is, of course, an important one in a practical application, but the field of cryptography is fairly well defined and need not be addressed here as a subject of our research interest.
- Third, we assume satisfactory protection of the circuits of communication to insure useful information is not captured by unintended parties. In reality, this type of protection is commonly built into a system transmitting classified data and is mostly a physical protection issue which is of little interest here.

This paper addresses the implementation of non-discretionary security control mechanisms in a distributed system. Lampson's reference monitor is extended from a single computing node to a networked environment where the sum of all computing nodes constitute the distributed reference monitor that we choose to call a sentinel. The security control mechanisms could be based upon the Bell and LaPadula model, the SRI model, or any other. The ideas presented are intended to be free from a particular security model and are therefore general. Examples are provided using the Bell and LaPadula model only because it is the most widely stated in current literature and most in use throughout the community.

2. Background

The provision of security and privacy in computer systems is not a new concern nor is its requirement confined to those applications of a government or military nature. Its applicability is universal throughout all systems whose continued existence depends upon the ability to reasonably safeguard and segregate information according to some model of user desire (e.g., electronic funds transfer, corporate planning data, unannounced product designs, etc.). The case for a computer security need in the "business systems" world is fast being discovered. From 1967 through 1970, the Department of Defense sponsored studies which addressed safeguards in remote access, resource sharing computer systems. These studies furnished the necessary catalyst for significant research findings and events. The more notable of these are Lampson's reference monitor, Schell's security kernel, the Bell and LaPadula security model, and the formulation of the DoD Computer Security Center. This latter entity was given the charter to encourage the widespread availability of trusted computer systems and to evaluate such systems for use within the DoD. To this end, it published DOD-5200-28-STD [3] which stands today as the primary guidance to software suppliers in obtaining evaluation of their trusted computing products, and more importantly, it defined several distinct evaluation classes of progressively increasing confidence (from a low of class "D" to a high of class "A1"). Trusted computing base software begins at the B2 level.

Systems processing classified data generally can be partitioned into one of three modes of operation [2]:

(1) Dedicated mode. The system processes data of a single classification or compartment only. All users are cleared and/or approved for that single level of access to the information involved. All programs and data have a security classification associated with them and appropriate security labeling of all output must be assured.

(2) System high mode. This differs from the dedicated mode in that the system supports more

tributed systems environment will be examined. In discussing the extension of the reference monitor abstraction into distributed systems, certain assumptions need to be made to avoid confusing the problem with less germane issues. Our assumptions are listed below:

- First, we assume adequate, acceptable physical protection of each of the distributed system components and their attached peripheral devices. This is considered a reasonable assumption and one that would command a high level of attention in a practical implementation.
- Second, we assume the use of adequate and appropriate cryptography which will insure the absolute secrecy and privacy of the information transmitted from the physical protection boundary of the sending component to the protection boundary of the receiving component. We are not concerned with the method of information relay or the type of cipher mechanism used. This issue is, of course, an important one in a practical application, but the field of cryptography is fairly well defined and need not be addressed here as a subject of our research interest.
- Third, we assume satisfactory protection of the circuits of communication to insure useful information is not captured by unintended parties. In reality, this type of protection is commonly built into a system transmitting classified data and is mostly a physical protection issue which is of little interest here.

This paper addresses the implementation of non-discretionary security control mechanisms in a distributed system. Lampson's reference monitor is extended from a single computing node to a networked environment where the sum of all computing nodes constitute the distributed reference monitor that we choose to call a sentinel. The security control mechanisms could be based upon the Bell and LaPadula model, the SRI model, or any other. The ideas presented are intended to be free from a particular security model and are therefore general. Examples are provided using the Bell and LaPadula model only because it is the most widely stated in current literature and most in use throughout the community.

2. Background

The provision of security and privacy in computer systems is not a new concern nor is its requirement confined to those applications of a government or military nature. Its applicability is universal throughout all systems whose continued existence depends upon the ability to reasonably safeguard and segregate information according to some model of user desire (e.g., electronic funds transfer, corporate planning data, unannounced product designs, etc.). The case for a computer security need in the "business systems" world is fast being discovered. From 1967 through 1970, the Department of Defense sponsored studies which addressed safeguards in remote access, resource sharing computer systems. These studies furnished the necessary catalyst for significant research findings and events. The more notable of these are Lampson's reference monitor, Schell's security kernel, the Bell and LaPadula security model, and the formulation of the DoD Computer Security Center. This latter entity was given the charter to encourage the widespread availability of trusted computer systems and to evaluate such systems for use within the DoD. To this end, it published DOD-5200-28-STD [3] which stands today as the primary guidance to software suppliers in obtaining evaluation of their trusted computing products, and more importantly, it defined several distinct evaluation classes of progressively increasing confidence (from a low of class "D" to a high of class "A1"). Trusted computing base software begins at the B2 level.

Systems processing classified data generally can be partitioned into one of three modes of operation [2]:

(1) Dedicated mode. The system processes data of a single classification or compartment only. All users are cleared and/or approved for that single level of access to the information involved. All programs and data have a security classification associated with them and appropriate security labeling of all output must be assured.

(2) System high mode. This differs from the dedicated mode in that the system supports more

than one classification or compartment of information. Some need-to-know controls may be implemented, but regardless, all users must be cleared and/or approved for access to all levels of information supported by the system. As in the dedicated mode, all programs and data have a security classification associated with them and appropriate labeling of all output must be assured. Output is then "manually" reviewed to insure correct labeling. This mode of operation is characterized by its high cost disadvantage in terms of personnel clearances and manual interventions.

(3) Multilevel mode. The principal characteristic here is that not all users are cleared and/or approved to the same level of information access and the system is trusted to separate users from data they are not authorized to see by security classification and need-to-know. Security-type labels are associated with each data object and subject, and access is mediated by a reference monitor mechanism.

In this paper, we will concentrate on Multilevel mode ((3) above) in trying to develop the reference monitor extension to the distributed level of operation. Once that is accomplished, it becomes necessary to again examine what happens to the reference monitor at the distributed level if the mode of operation of a component system changes to mode 1 or mode 2. These issues and others are addressed below in an abbreviated form. A more comprehensive treatment can be found in [15].

Finally, we state the motivation for this paper. Single system operation is fast giving way as a norm to networked distributed system implementations. This has occurred as a natural evolutionary process which has been accelerated by the rapidly declining costs of both hardware and communications coupled with rapid advances in distributed software reliability. Furthermore, we have seen a proliferation of the small computer into the office place and the emergence of new disciplines such as the Office Information Systems (OIS). The same concerns and issues of security that are present in traditional computing environments are beginning to surface within OIS as a serious research topic [14]. The ideas expressed in this paper are applicable to the interconnected office systems as well as to the more traditional network. We believe that the reference monitor concept can be readily adapted to this new environment by realizing its abstraction through a different approach.

3. Discretionary vs. non-discretionary access controls

Access controls to information objects in an automated system can be divided into two categories, discretionary and non-discretionary. At the lower end of security control, we find a need for the discretionary control. Discretionary access control is a means of restricting access to objects based on the identity of subjects and/or groups to which they belong. The controls are discretionary in the sense that a subject with a certain access permission is capable of passing that permission (directly or indirectly) on to any other subject. A comprehensive treatment of this topic can be found in NCSC-TG-003 [4]. Non-discretion access controls (also known as mandatory access controls) are a means of restricting access to objects based on the sensitivity of the information contained in the objects and the formal authorization (e.g., security clearance) of the subjects to access information of such sensitivity. Non-discretionary controls are used to implement a desired model of security and are typically enforced by the operating system. If a conflict exists between a subject's desire to grant access to a system object and the denial of that request under the security rules enforced by non-discretionary controls, the non-discretionary controls take precedence and access is not granted. The enforcement of security rules by mandatory controls must always have the higher priority between the two access controls.

4. The reference monitor

Early in secure operating system research it became clear that modification of existing operat-

ing systems to provide a secure computing base was an inappropriate approach due to the size and complexity of the operating system and the expense of its verification [12].

The concept of a reference monitor was introduced by Lampson [8] as an abstract notion designed to mediate access to passive *objects* within a system (files, storage devices, utility programs, etc.) by active *subjects* (users, processes). This mediation is accomplished by accessing some stored, trusted set of rules which implements a particular security model. This concept is depicted in Fig. 1.

Three key engineering concepts, *mediation, isolation*, and *verification* must exist in a reference monitor implementation. The first is that of mediation, meaning that all references by untrusted subjects to untrusted objects must be controlled by the trusted reference monitor. Second, the reference monitor itself must be isolated (separated) from access by untrusted software. Third, some formal and/or informal methodology must be used to prove correspondence between the reference monitor and the security model it implements. The purpose of this third component is to show faithful implementation of the mathematical model of security rules. A step-wise approach to this verification process is given in Ref. [1].

In practical applications, the reference monitor abstraction has been implemented through the use of Schell's security kernel first introduced in 1972, and tested successfully in 1974, by the MITRE Corporation. This idea fits well with the

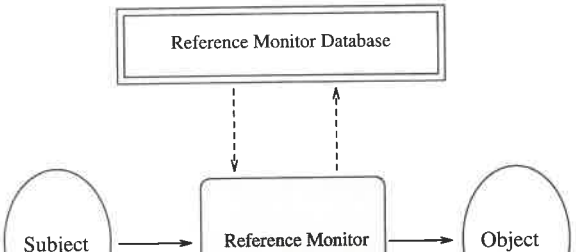

Fig. 1. Reference monitor.

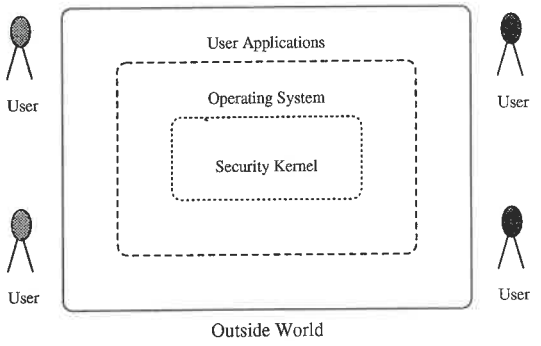

Fig. 2. Embedded security kernel.

notion of kernelized operating systems and protection domains. The security kernel is trusted software that runs in a protection domain higher than that of the operating system. It is embedded as a kernel within the innermost layer of the operating system and implements all the security relevant operations in the system [9]. This structure is shown in Fig. 2. The objectives of the kernel are threefold. First, it provides *mediation* between active subjects and passive objects within the system. If, for example, a process wishes to access a specific file, the file request is trapped by the operating system, passed to the security kernel, checked against the security model rules base, and then is either allowed or disallowed. The second objective, *isolation*, is met by the inherent nature of the kernel design and implementation of protection domains. Finally, the third objective, *verification*, is made considerably easier to accomplish by virtue of the limited size of the kernel. Since only that software necessary to enforce the security model is incorporated within the kernel, the amount of software to be verified is considerably reduced over that for a full-sized operating system. The difficulty then becomes one of choosing which operating system routines need to be incorporated in the kernel while trying to keep the kernel size as small as possible.

It is important to note that of the three security models of operation (dedicated, system high, and multilevel), only multilevel requires implementation of a reference monitor. In a dedicated system only one classification of object exists and therefore no mediation is required. In a system

high mode of operation, all users have a clearance equal to the highest level of information stored, so trusted software is not required (although some mediation may be enforced by the operating system). Manual checking of the system output is required to insure that the system properly labels the data with its correct classification.

The final point we make concerning reference monitors is that the implementation of the monitor database must be done in trusted software. Rules captured in such a database typically implement some agreed upon model such as the Bell and LaPadula model or perhaps some form of an information flow model for non-discretionary access control. Discretionary access may be controlled in the same database by implementing a capability list or access control list. Subjects and objects active in the system will each have a security clearance tag associated with them for access by the reference monitor. Obviously, this tag must be non-forgeable and inalterable. Successful commercial implementations of the above approach include Honeywell's SCOMP and Multics systems.

5. Distributed systems considerations

Until this point we have discussed the workings of a reference monitor in a single system environment and the engineering considerations that must be present. The distributed environment presents quite a different problem. Here we have two or more systems, separated by some geographical distance, which are cooperating with each other to perform some common task at hand. There appears to be no commonly accepted definition of distributed systems and the term is often used interchangeably with computer networks. A distributed system can be defined as having a systemwide operating system where, to the user, the entire collection of processors appears as one. The view expressed by some other authors is less restrictive as they say, "A distributed system is one in which the computing functions are dispersed among several physical computing elements" [13]. We chose to accept the latter description so as to include more systems. It is interesting to note that this latter definition also includes the interconnected OIS environment whereas the former may not.

A solution to providing a multilevel secure distributed environment is presented in Ref. [12]. Here the authors claim that the security kernel (reference monitor) approach is likely to be unaffordably slow by a factor of three to ten over a conventional system. Furthermore, they describe an architecture known as the Distributed Secure System (DSS) that is largely based upon hardware and firmware modification to an existing UNIX network. The solution seems to be somewhat limited and hardware laden. Although their solution does appear to be workable, it seems prudent to further examine the extension of a reference monitor to the distributed level of a system.

We believe it is necessary to realize the reference monitor abstraction in two different ways, corresponding to the two levels present within a distributed environment. Within the single system, the abstraction can be realized through the implementation of a security kernel as described above. This kernel may be primarily a software feature with hardware support. When we move to the distributed level, we encounter the additional problems of data transmission, participating node communication, and node trust. Furthermore, various distributed architectures can impact the reference monitor's responsibility. For example, if the distributed system is a collection of systems each with their own full operating system capability which includes an embedded security kernel, then the reference monitor abstraction at the distributed level can be implemented primarily with network interfaces and appropriate cryptographic devices. On the other end of the spectrum, if the system is implemented with a system-wide operating system, the task at hand becomes far more complicated and may require more of a software solution at the network level in addition to cryptographic and gateway protections. Beyond the network topology, an additional concern we have is the mode of operation for each participating node. If all nodes are running at the multilevel secure level, we say we have a *homogeneous modus operandi* of cooperating trusted hosts. If on the other hand, we have a

heterogeneous modus operandi, we may not be able to form a secure distributed system due to lack of trust amongst the users. These problems and others are discussed below. The ideas for the approaches that follow were gleaned from many sources, but the more helpful and most recent were Refs. [2,10,16].

We refer to the entity connected to the network as a "node". Admittedly, this is a rather loose usage of the term, but we desire to encompass the traditional mainframe at one end of the spectrum, as well as an OIS (with many computer-like devices communicating with each other across a LAN) at the opposite end. We believe our suggestions are general enough to include the entire range of systems.

Let us first examine the distributed system where each participating node has a security kernel and is certified multilevel secure (MLS). Furthermore, we assume here that the MLS systems all have the same upper bound of authorized clearance (e.g., top secret) and that all implement the same security model (e.g., Bell and LaPadula). Consider the system depicted in Fig. 3. Here we can trust each participating node so our primary concern is the protection of the data while in transit and its delivery to the correct destination. The protection required for such a system is shown in Fig. 3 by concentric circles. The outermost layer represents some recognized, acceptable (to the user) encryption/decryption method common to all nodes in the system.

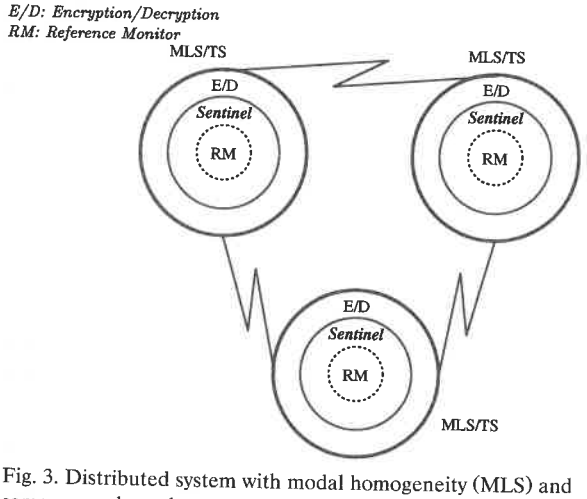

Fig. 3. Distributed system with modal homogeneity (MLS) and same upper bound.

We need not be concerned here with physically different encryption schemes for the various levels of information and can trust the sentinel and reference monitor (RM) to provide the necessary separation. The encryption/decryption (E/D) provides the isolation of the data from the outside world that we require. Data or messages arriving at a node are tobe decrypted and passed in clear text to the sentinel. The sentinel is responsible for insuring that the correct delivery takes place (node-wise) and that the arriving data object is properly tagged with its security classification. It is then the responsibility of the sentinel to pass the object to the RM in a manner that makes the sentinel appear as simply another user process. We desire that the RM mediate this request as it would any other and respond accordingly whether access is allowed or not. Requests for objects not local to a particular node are to be passed to the sentinel where a check occurs to insure the subject's security clearance is present and that appropriate routing information is appended. The sentinel then passes the request to the E/D for encryption prior to forwarding.

The problem changes somewhat if we add a node that does not exhibit a MLS upper bound equal to others in the system. If, for example, we add a node that is certified to operate at a MLS-Secret (MLS/S) level upper bound to an existing system that is MLS-Top Secret (MLS/TS), we must insure that the sentinel prohibits a violation of the security model being implemented. If that model is based on the Bell and LaPadula rules then, for example, we would want to permit the secret node to accept requests from subjects labeled Secret or below, but to prohibit the same requests from subjects labeled Top Secret, thereby enforcing the "*no write down rule*" of the $*$-property. Likewise, we would permit the new node to access any other system node since they are all trusted and can be relied upon to separate Top Secret material from view by the Secret node or its agents. Fig. 4 shows such a configuration. Now the sentinel must take on more responsibility in screening incoming requests for objects to insure they can be properly

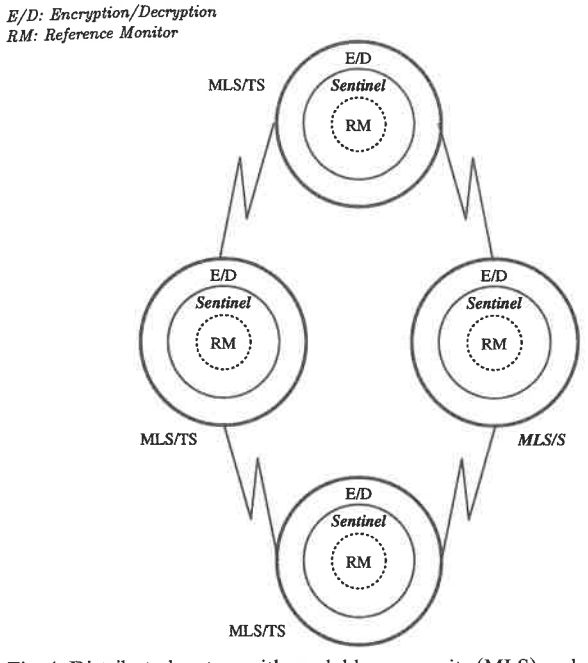

Fig. 4. Distributed system with modal homogeneity (MLS) and different upper bounds.

served by the node. Requests from Top Secret subjects must be rejected by the sentinel located at the MLS/S node. This sentinel need not be concerned with the level of information requested by its subjects since the reference monitor of the receiving node will be responsible for segregation [1].

In the above examples, we viewed the distributed systems as all having MLS nodes. We then added an additional layer of software known as the sentinel to each node. The reference monitor abstraction for such systems can then be realized by the summation of all the sentinel, over the entire system. If we now allow violation of the MLS property by any participating node, the security status of our system changes dramatically. This situation cannot be permitted without additional changes in the sentinel software/firmware implementation which are discussed below. We wish to note at this point that this situation would be expected to arise frequently with any rapidly expanding network. In particular, we would expect to find the desired interconnection of OIS between different organizations in the commercial world to involve this very aspect and would argue for centralized control of sentinel access and settings.

When we drop below the MLS mode of operation in a security conscious system, we assume only two other permissible modes – system high or dedicated. It is important to note here that both these modes are untrusted and place a high reliance upon factors outside the system to assure the security of this operation. We propose to examine each of these modes separately.

5.1. System high mode considerations

First, we consider the system high (SH) mode. Simply stated, the participation of a system high node forces the security system view to consider that node as only containing objects and subjects at the highest security level permitted by the system. The portion of the sentinel supporting this node must be trusted to label every outgoing subject or object with the highest classification of information maintained by the system regardless of the classification assigned within the system itself. Likewise, every request for information arriving at this node must be considered a request for information at the highest level allowed within the node. The sentinel must be trusted to accomplish these functions since the local software cannot be trusted to separate objects and subjects according to clearances. Isolation of the sentinel could be incorporated in software by creating a protection layer, but it is more likely a candidate for firmware implementation. If this approach is

[1] It is true that with this approach, in an interconnected MLS environment, the sentinel must be trusted to reject an incoming request if it violates the security model. This does not necessarily invalidate the technique for inter-organizational networks. It just means that practical agreements must be reached between organizations that assure common controls. These can be accomplished by mutual inspections, third party contracts, or mutual support agreements. If one uses a technique that calls for the sentinel to only be as trusted as the local node – we introduce a new problem, that of coming up with an algebra to determine risk associated with linking together sentinels with different assurances. Both approaches can be made to work, but both have shortcomings.

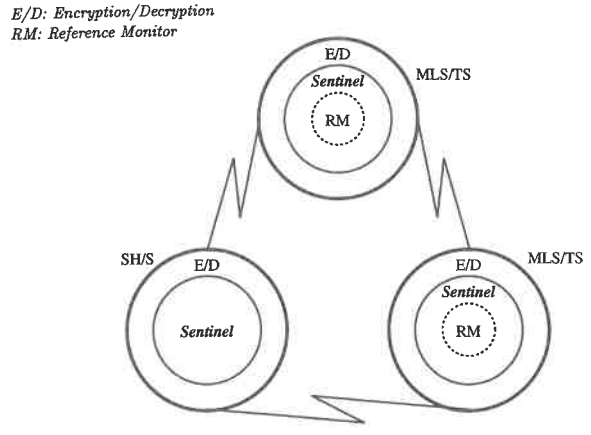

Fig. 5. Distributed system with heterogeneous modus operandi.

selected, isolation can be accomplished by physical separation of the board from other components in the system. The node itself will not include a reference monitor since it does not operate at a multi-level mode of operation as shown in Fig. 5.

5.2. Dedicated mode considerations

If the distributed system consists of dedicated or system high nodes only, the reference monitor concept need not be applied since the systems themselves provide separation of data by classification. Isolation can best be achieved here by creating a series of separate front end machines to insure that the security model is enforced throughout the system. Similarly, different cryptographic keys for different security classifications might be helpful. Such a solution is proposed in Ref. [12] and will not be further discussed in this paper.

The preceding configurations outline some likely distributed systems and demonstrate how security can be enforced. It is important to understand that in each case the reference monitor was applied from the top down. First we examine the security mode of each component of the system, then we insure the isolation and separation characteristics. If the mode of operation changes for a single participating node, or if a new node is added to the system, the entire security status of the overall system changes and must be again viewed from the top down with appropriate hardware, software, and firmware changes made. The single most important ingredient is the element of trust. If the processors are trusted (MLS) they can be relied upon to properly report the security level of their objects or subjects, but if they are not trusted, then every object or subject must be assumed to be of the highest level of security classification existing on that processor and treated accordingly.

6. Conclusions

The concept of a reference monitor as proposed by Lampson can in fact be extended to the distributed environment and is most likely useful in the emerging OIS arena. It can be distributed over participating nodes in a system where the sum of all its pieces constitutes the distributed reference monitor that we choose to call a sentinel. The functions of the sentinel change depending upon the mode of operation of the node it supports. The major difference between implementing a single system reference monitor (RM) and a sentinel is in the way we provide for mediation and isolation. A reference monitor normally provides for mediation by software controls and isolation by security kernel implementation within the operating system. The sentinel is forced to include cryptography and firmware to insure mediation and isolation. Furthermore, a RM only exists in MLS systems whereas a sentinel may be required in all three modes (MLS, system high, and dedicated). The major decisions on how to implement the sentinel are based upon the trust inherent in the node it exists on. This paper did not address the specifics of how a sentinel would work at each different case or what specific rules would be enforced when viewed as a state machine. The specifics have been presented in a very detailed fashion in Ref. [15] and should the reader be interested in further exploring the work, the reference is easily accessible. A similar technique has been employed with both the *BLACKER* and *CANEWARE* devices developed in recent years. Areas we have not examined that

might prove fruitful include limiting the topology of the network to one that supports mediation (e.g., a Star) and, secondly, the implementation of trusted, kernelized object servers throughout the distributed system that isolate and mediate all requests. Additionally, we did not explore the impact of connecting multi-level secure systems together, each with the same upper bound but with different levels of assurance as defined by [3], e.g., AB2 evaluated system connected to a B3 or A1. Each of these areas would make interesting papers in themselves.

References

[1] S.R. Ames, M. Gasser and R.R. Schell, Security kernel design and implementation: An introduction, *Computer* **16**(7) (1983) 15–21.

[2] J. Anderson, A unification of computer and network security concepts, in: *Proc. of the IEEE 1985 Symposium on Security and Privacy*, pp. 77–87, 1985.

[3] DoD, Department of Defense Trusted Computer System Evaluation Criteria, Department of Defense Computer Security Center, Fort George G. Meade, Md. 20755, August 1985.

[4] DoD, A Guide to Understanding Discretionary Access Controls in Trusted Systems, National Computer Security Center, Fort George G. Meade, Md. 20755, August 1987.

[5] D. Estrin, Non-discretionary controls for inter-organization networks, in: *Proc. of IEEE Symposium on Security and Privacy*, pp. 56–61, 1985.

[6] D. Estrin, Controls for interorganization networks, *IEEE Trans. Software Eng.* **13**(2) (1987) 249–261.

[7] P. Janson and R. Molva, Security on open networks and distributed systems, *Computer Networks and ISDN Systems* **22**(5) (1991) 323–346.

[8] B.W. Lampson, Protection, in: *Proc. Fifth Princeton Symposium on Information, Sciences and Systems*, pp. 437–443, March 1971.

[9] C.E. Landwehr, Formal models for computer security, *Computing Surveys* **13**(3) (1987) 247–278.

[10] D. Nessett, Factors affecting distributed system security, *IEEE Trans. Software Eng.* **13**(2) (1987) 233–248.

[11] J.K. Reynolds, The helminthiasis of the Internet, *Computer Networks and ISDN Systems* **22**(5) (1991) 347–361.

[12] J.M. Rushby and B. Randall, A distributed secure system, *Computer* **16**(7) (1983) 55–67.

[13] A. Tanenbaum, *Computer Networks* (Prentice-Hall, second edition, 1988).

[14] R. Vaughn, H. Saiedian and E. Unger, A survey of security issues in office computation and the application of secure computing models to office systems, *Comput. Security* **12**(1) (1993) 79–97.

[15] R.B. Vaughn, A security architecture for office automation systems, PhD thesis, Kansas State University, 1988.

[16] S.T. Walker, Network security overview, in: *IEEE Symposium on Security and Privacy*, pp. 62–76, 1985.

Rayford B. Vaughn, Jr. received his Ph.D. in Computer Science from Kansas State University in 1988. He is currently the Commander of the U.S. Army Information Systems Software Center at Fort Belvoir, Virginia, and has previously served with the National Computer Security Center and at the Assistant Department of the Army Information Manager. During the academic year 1990–1991, he served a one-year appointment as a Visiting Professor of Computer Science at the U.S. Naval Avademy, Annapolis, Marylands. Col. Vaughn is a member of the Armed Forces communications and Electronics Association. His research area includes security of office automation systems.

Hossein Saiedian is currently an Assistant Professor of Computer Science at the University of Nebraska at Omaha. He received his Ph.D. in Computer and Information Sciences from Kansas State University in 1989. Dr. Saiedian has over 40 technical articles in computer and information sciences journals and proceedings including articles in recent issues of *IEEE Computer Journal of Systems and Software, Journal of Software and Information Technology, Computer & Security, Journal of Microcomputer Applications, Office systems Research Journal*, and *Journal of Computer Science Education*. His research interest include formal methods, object-oriented computing, and office information systems. Dr. Saiedian is a member of the IEEE-CS, ACM, ACM SIGOIS, and is currently Chair of the ACM SIGICE.

Elizabeth A. Unger is Professor of Computing and Information Sciences at Kansas State University. Her areas of research interest include database systems and the use of the object oriented approach to produce sound systems that enforce integrity and security. The security of database management systems is her primary research focus with emphasis on inferential security and integrity. Prioir to earning a doctorate in 1978 she had a career in the management of computing service centers.