MULTISAFE--A DATA SECURITY ARCHITECTURE

by

Robert P. Trueblood

H. Rex Hartson*

Department of Computer Science
University of South Carolina
Columbia, South Carolina 29208

INTRODUCTION

MULTISAFE is a MULTI-module system for supporting Secure Authorizations with Full Enforcement for database management. The architecture of MULTISAFE combines new ideas with existing approaches such as "back-end" computers [CANAR74], "encapsulation" [BISBR74], object program division [LANGT76], and "user virtual machines" [COOKT75]. The MULTISAFE architecture provides a framework for seeking solutions to data security problems such as limited ability and variety of security mechanisms, verification of security mechanisms, and system performance overhead.

BASIC ARCHITECTURE

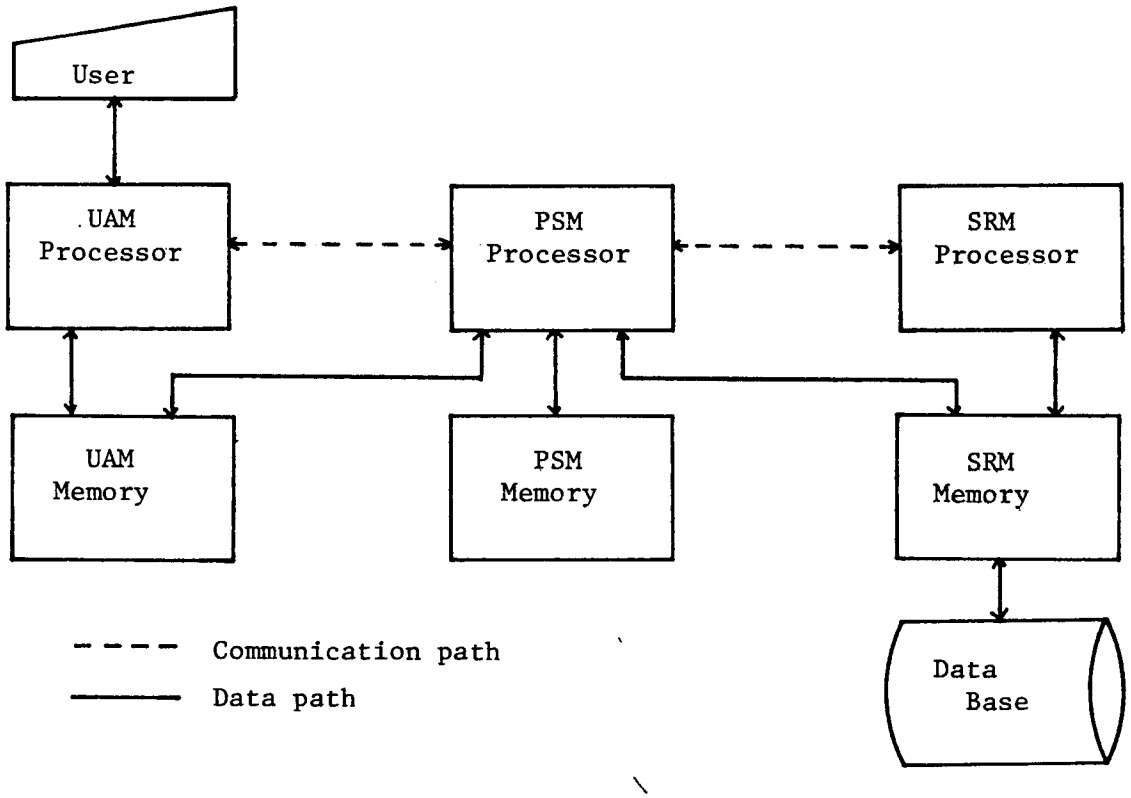
The MULTISAFE architecture is composed of three modules: (1) the User and Application Module (UAM), (2) the Protection and Security Module (PSM), and (3) the Storage and Retrieval

---

* Address: Dept. of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061.

Module (SRM). The UAM can support many different types of user interfaces. The SRM can support any of the three data models (relational, network, and hierarchical), or it can be a specialized database machine. MULTISAFE can be implemented on one or more processors. In an ordinary single processor system these three modules function sequentially in an interleaved fashion. In a multiprocessor system all three modules can function in a concurrent fashion. The UAM coordinates and analyzes user requests concurrently with the SRM's generation of database responses for previous requests. Simultaneously, the PSM performs security checks on the system's activities. In the remainder of this note the multiprocessor view is used.

The basic multiprocessor architecture for MULTISAFE is illustrated in the figure below. Three separate processors are



Basic Architecture

connected to three separate primary random access memories, utilizing a multiport, private memory organization [ENSLP77]. This organization permits the PSM processor to access (read/write) all three memories while the UAM and SRM processors are restricted to their own memories. The effect is module isolation by physical boundaries.

FRAMEWORK FOR SOLUTIONS

The MULTISAFE architecture provides a useful framework for formulating solutions to the three data security problems in the following manner. Solutions for the first problem, limited ability and variety of security mechanisms, are fostered by the provision of a dedicated processor and memory for efficiently implementing and executing sophisticated mechanisms. The PSM can support numerous security mechanisms and techniques such as formularies [HOFFL71], or predicate-based models of protection [HARTH7], as well as encryption, history keeping, security auditing, and inference controls.

The physical isolation of modules described earlier constrains all intermodule communication to well-defined channels and eliminates all "back door" access paths in the software. As a consequence, security depends upon the correctness of the mechanisms within the PSM and upon the correctness of intermodule communication. Verifying the correctness of the authorization and enforcement processes of the PSM can now be isolated as a separate endeavor. Currently available procedures can then be applied to show that the specifications of these processes do not violate protection policies (axioms) and that the programs are

faithful to the specifications.

The correctness verification of intermodule communication, in the general case, can be made more difficult by the addition of process concurrency. However, the architecture of MULTISAFE has been designed with an overriding concern for the simplicity of intermodule communication, because of its importance to security. First, coupling among the modules is extremely loose. This eliminates the concerns about synchronization which occur, for example, with semaphores and similar interprocess signals.

Next, all three modules operate quite independently, each being driven by request and response messages in its input queue. When, in order to continue with its current process, a module A needs a function performed by module B, module A puts its process to sleep, sends a request message to module B, and continues with other work. Eventually, B will select the message from its queue, compute the function, and respond into A's input queue. When A finishes its current task and looks in its queue for another one, it will find the results for B's function. A can then reinstate the original process and continue its processing. Therefore, the order of messages in the queue is not important with regard to correctness or synchronization. Correctness of inter-module communication depends on three very simple processes: a control loop (for selecting items from a queue), a sending loop, and a receiving loop. The important parts of each of these processes require a page or less of straightforward code. The messages are encapsulated as abstract data types to maintain their integrity and security. Further, all three MULTISAFE modules use the same set of inter-module communication

programs, so that verification needs to be done only once for the whole system.

The procedures have been designed and specified in [TRUER81]. The verification step was not taken; program verification is not part of this research. However, the methodology for doing this verification is now well established elsewhere in the literature.

In some systems, performance overhead, the subject of the third problem area, can be as low as 10% or less. However, it has been reported in [WOODF74] that meticulous data-dependent security checking can impose up to a 200% performance penalty. The MULTISAFE architecture counteracts such burdensome performance degradation through concurrent processing of the three modules. In addition, the communication bottlenecking typically found in multiprocessor systems is overcome by dividing intermodule messages into two parts: a fixed length message descriptor and a variable length message text. Message descriptors flow from processor to processor while message text flows directly from memory to memory. A simulation study of MULTISAFE performance is given in [TRUER80], and a analytic performance model is being studied at USC. Some work is also presently being conducted at VPI & SU on cost functions for MULTISAFE protection performance analysis.

ACKNOWLEDGEMENT

The authors gratefully acknowledge contributions to MULTISAFE by Johannes J. Martin of VPI & SU.

REFERENCES

BISBR74 Bisbey II, Richard L., and Gerald J. Popek, "Encapsulation: An Approach to Operating System Security," Proceedings of the ACM Annual Conference, San Diego, Ca., (November 1974), pp. 666-675.

CANAR74 Canaday, R. H., R. D. Harrison, E. L. Ivie, J. L. Ryder, and L. A. Wehr, "A Back-end Computer for Data Base Management," CACM, Vol. 17, No. 10, (October 1974), pp. 575-582.

COOKT75 Cook, Thomas J., "A Data Base Management System Design Philosophy," ACM SIGMOD International Conference on Management of Data, (May 14-16, 1975), pp. 15-22.

ENSLP77 Enslow, P. H., "Multiprocessor Organization--A Survey," ACM Computing Surveys, Vol. 9, No. 1, (March 1977), pp. 103-129.

HARTH76 Hartson, H. Rex, and David K. Hsiao, "A Semantic Model for Data Base Protection Languages," Proceedings of the International Conference on Very Large Data Bases, Brussels, Belgium, (September 1976).

HOFFL71 Hoffman, Lance J., "The Formulary Model for Flexible Privacy and Access Control," Proceedings of the Fall Joint Computer Conference, Vol. 39, (1971), pp. 587-601.

LANGT76 Lang, T., E. B. Fernandez, and R. C. Summers, "A System Architecture for Compile-time Actions in Databases," IBM Los Angeles Scientific Center, Report No. G320-2682, (December 1976), pp. 1-26.

TRUER80 Trueblood, Robert P., "Design Extensions and Feasibility Study of MULTISAFE," Summer I 1980 College Research Grant Report, Tech. Report No. TR80002, Department of Computer Science, University of South Carolina, Columbia, S.C. 29208.

TRUER81 Trueblood, Robert, H. Rex Hartson, and Johannes J. Martin, "MULTISAFE--A Modular Multiprocessing Approach to Secure Database Management," (submitted for publication, 1981).

WOODF74 Woodward, Franklin G., and Lance J. Hoffman, "Worst Case Costs for Dynamic Data Element Security Decisions," Proc. of the ACM Annual Conf., San Diego, Ca., (November 1974), pp.539-544.