



Calhoun: The NPS Institutional Archive

Theses and Dissertations

Thesis Collection

1999-09

**Framework for a high-assurance security extension
to commercial network clients**

Balmer, Steven R.

Monterey, California. Naval Postgraduate School

<http://hdl.handle.net/10945/13657>



Calhoun is a project of the Dudley Knox Library at NPS, furthering the precepts and goals of open government and government transparency. All information contained herein has been approved for release by the NPS Public Affairs Officer.

**Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943**

<http://www.nps.edu/library>

NAVAL POSTGRADUATE SCHOOL MONTEREY, CALIFORNIA



THESIS

**FRAMEWORK FOR A HIGH-ASSURANCE
SECURITY EXTENSION TO COMMERCIAL
NETWORK CLIENTS**

by

Steven R. Balmer

September 1999

Thesis Advisor:
Second Reader:

Cynthia E. Irvine
James P. Anderson

Approved for public release; distribution is unlimited.

DTIC QUALITY INSPECTED 3

20000313 032

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE September 1999	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE FRAMEWORK FOR A HIGH-ASSURANCE SECURITY EXTENSION TO COMMERCIAL NETWORK CLIENTS		5. FUNDING NUMBERS	
6. AUTHOR(S) Steven R. Balmer		8. PERFORMING ORGANIZATION REPORT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.	
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.		12b. DISTRIBUTION CODE	
13. ABSTRACT (maximum 200 words) The Department of Defense and U.S. Government have an identified need to securely share information classified at differing security levels. Because there exist no commercial solutions to this problem, NPS is developing a Multilevel Secure Local Area Network (MLS LAN). The MLS LAN extends the high assurance capabilities of an evaluated multilevel secure system to commercial personal computers (PCs) running commercial operating systems and office productivity software by using a Trusted Computing Base Extension (TCBE). The TCBE is intended to provide trusted path and object reuse supporting services to the network TCB. This thesis describes the physical interfaces required for the TCBE to complete a trusted path and control the client PC. Potential implementations for each interface are suggested and analyzed for security implications. Also presented is a detailed analysis of methods for delivering the Windows NT operating system (including the suitability of Terminal Server Edition) to the client PC in the MLS LAN with high assurance of properly controlled object reuse and operating system integrity.			
SUBJECT TERMS Multilevel Security, Trusted Path, High-Assurance, Network Client		15. NUMBER OF PAGES 118	16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. UL

THIS PAGE INTENTIONALLY LEFT BLANK.

Approved for public release; distribution is unlimited.

**FRAMEWORK FOR A HIGH-ASSURANCE SECURITY
EXTENSION TO COMMERCIAL NETWORK CLIENTS**

Steven R. Balmer
Lieutenant, United States Navy
B.S.E.E., University of New Mexico, 1991

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

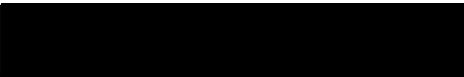
from the


**NAVAL POSTGRADUATE SCHOOL
September 1999**


Author:


Steven R. Balmer

Approved by:


Cynthia E. Irvine, Thesis Advisor


James P. Anderson, Second Reader


Dan Boger, Chairman
Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK.

ABSTRACT

The Department of Defense and U.S. Government have an identified need to securely share information classified at differing security levels. Because there exist no commercial solutions to this problem, NPS is developing a Multilevel Secure Local Area Network (MLS LAN). The MLS LAN extends the high assurance capabilities of an evaluated multilevel secure system to commercial personal computers (PCs) running commercial operating systems and office productivity software by using a Trusted Computing Base Extension (TCBE). The TCBE is intended to provide trusted path and object reuse supporting services to the network TCB.

This thesis describes the physical interfaces required for the TCBE to complete a trusted path and control the client PC. Potential implementations for each interface are suggested and analyzed for security implications. Also presented is a detailed analysis of methods for delivering the Windows NT operating system (including the suitability of Terminal Server Edition) to the client PC in the MLS LAN with high assurance of properly controlled object reuse and operating system integrity.

THIS PAGE INTENTIONALLY LEFT BLANK.

TABLE OF CONTENTS

I. INTRODUCTION.....	1
A. BACKGROUND.....	1
B. GOALS/THESIS QUESTIONS	4
1. Trusted Computing Base Extension (TCBE).....	9
C. ORGANIZATION OF THESIS	13
II. FUNCTIONAL DESIGN REQUIREMENTS FOR A TRUSTED COMPUTING BASE EXTENSION.....	15
A. PURPOSE OF THE TCBE.....	15
1. Trusted Path Communications	16
2. Object Reuse	18
B. TCBE EXTERNAL INTERFACES.....	19
1. Background.....	20
2. User Input to the TCBE.....	22
3. TCBE Video Display to the User.....	24
4. TCBE Interface with the Personal Computer.....	27
5. TCBE Interface with the High Assurance Server (XTS-300)	30
6. TCBE Hard Drive Interface	33
C. CONCLUSIONS AND OBERVATIONS FROM THE EXTERNAL INTERFACE ANALYSIS.	37
III. DISK BASED OPERATING SYSTEM DELIVERY METHODS	41
A. A STATEMENT OF THE PROBLEM.....	41
B. POSSIBLE OPERATING SYSTEM DELIVERY METHODS.....	41
1. Removable Media.....	45
2. Large Permanent Disk with Multiple Partitions.....	49
3. Multiple Permanent Disks (One for Each Level).....	52
4. Multiple Permanent Disks with Multiple Partitions	56
5. Disk Mirroring/Ghosting Applied to a Single Permanent Disk.....	59
6. Disk Mirroring/Ghosting Applied to Multiple Disks	63
C. GENERAL COMMENTS AND CONCLUSIONS.....	66
IV. USING WINDOWS NT 4.0 TERMINAL SERVER EDITION	67

A. OVERVIEW OF WINDOWS NT 4.0 TERMINAL SERVER EDITION (TSE).....	67
B. TERMINAL SERVER EDITION TOPOLOGIES FOR THE MLS LAN	71
1. Case 1 – TSE on the LAN	72
2. Case 2 – TSE Under the HAS.....	75
3. Case 3 – TSE Enhanced with TCBE on the LAN.....	77
4. Case 4 – Multiple TSEs on LAN	79
5. Case 5 – Multiple Enhanced TSEs on LAN.....	81
6. Case 6 – An Ideal Solution.....	83
C. TSE GENERAL ANALYSIS AND CONCLUSIONS.....	85
V. CONCLUSIONS AND FUTURE WORK.....	87
A. RELATED WORK.....	87
1. Novell Trusted Workstation Partnership	87
2. Media Encryption Management System (MEMS) [Ref. 24].....	88
B. RECOMMENDATIONS FOR FUTURE RESEARCH	89
1. Trusted Paths/Channels Into Network Clients.....	89
2. Disk Ghosting Technology.....	90
3. Object Reuse in Commercial Personal Computers	90
4. BIOS Extensions in Trusted PCI Components.....	91
5. Trusted Hard Drive Encryption.....	91
6. Key Exchange Protocols	91
7. Identification and Authentication (I&A).....	92
8. Secure Bootstrap.....	92
C. CONCLUSIONS.....	92
APPENDIX A. INTEL 440BX MOTHERBOARD	95
A. THE PROCESSOR.....	95
B. BUSES	96
C. BIOS BASICS.....	97
APPENDIX B. SYMANTEC GHOST.....	99
LIST OF REFERENCES.....	101
INITIAL DISTRIBUTION LIST	103

LIST OF FIGURES

Figure 1. A Basic MLS LAN Architecture.....	8
Figure 2. Typical Intel Based Motherboard Showing Graphical Units	26
Figure 3. PCI Interrupt to IRQ Re-mapping.....	32
Figure 4. TCBE Functional Block Diagram	38
Figure 5. Operating System Delivery via Removable Media	45
Figure 6. Large Disk with Separate Partitions for Each Level.....	49
Figure 7. Separate Disks for Each Classification Level.....	52
Figure 8. Multiple Disks with Multiple Partitions	56
Figure 9. Ghosting Applied to a Single Disk with Multiple Partitions	59
Figure 10. Multiple Disks with Ghosting Technology	63
Figure 11. A Basic TSE LAN.....	70
Figure 12. An Enterprise-Level TSE LAN.....	71
Figure 13. TSE Server as a Peer on the MLS LAN	72
Figure 14. TSE and HAS in Series	75
Figure 15. Enhanced TSE on the LAN.....	77
Figure 16. Multiple TSEs for Separate Classifications.....	79
Figure 17. Enhanced TSEs at Different Classifications.....	81
Figure 18. TSEs Running on a Secure Virtual Machine Monitor	83
Figure 19. A Basic Block Diagram of an Intel 440BX Motherboard	95

THIS PAGE INTENTIONALLY LEFT BLANK.

ACKNOWLEDGMENT

I am eternally grateful to my God and Country for making this thesis possible. The support and encouragement given by my family made the experience one that I will never forget. Especially valuable were the input, support, and advice provided by my thesis advisors: Dr. Cynthia E. Irvine, and Mr. James P. Anderson. Their knowledge and helpfulness in every aspect of the MLS LAN project provided the insight and motivation to seek out the best solutions every step of the way.

THIS PAGE INTENTIONALLY LEFT BLANK.

I. INTRODUCTION

A. BACKGROUND

The handling and use of sensitive material is crucial to nearly every organization, government or civilian. The need to protect information regardless of the format, be it printed material, electronic documents, pictures, movies, audio, etc., is a reality that we all face. Not too long ago, people lived in a hardcopy world. Nearly every form of intellectual material was placed on printed-paper or some high-density media such as microfiche. Controlling this type of material is a straightforward task. Usually sensitive material of this sort is limited in distribution and has a strict written policy for its control with respect to who can view or copy it.

We are in the process of moving toward a paperless society that utilizes technologically advanced methods to distribute information. Nearly all forms of intellectual material are being moved into some electronic form. The storage devices have been interconnected to facilitate easy sharing and dissemination of information to those who have a need to know. Products that are available on the market for storing and delivering this information may not be appropriate. The Department of Defense (DoD) has particular requirements for the access and control of sensitive information. Motivation for this is the grave consequences to individuals and the threat to national security if that information is lost, compromised, or modified without authorization. [Ref. 1] Users of information technology products must have assurance that the information stored and accessed by these products remains private, available as needed, and without unauthorized modification. The hardware and firmware used to store information must handle the material properly so that improper dissemination, alteration, and loss do not occur.

The beginning of information assurance is the establishment of security policy. All organizations require a security policy, although not all organizations require the same security policy. A security policy is a set of rules that regulate how assets are managed, protected, and

distributed within a system or organization. Organizations control access to information using many criteria. Human access is determined by trust. Often users of systems are given access based on security clearances, background investigations, roles, or job-titles. This boils down to "a need to know." Users might be required to sign non-disclosure statements holding them legally responsible if information put under their trust is lost or compromised. The strength of this type of policy rests in the organization's ability to prosecute if such a loss is discovered. Organizational policies are often based on the type of material that they provide access to or by the job an individual must accomplish. Policies for these systems vary according to the level of concern the owners of the systems place on the value of the information that they contain. Segregation of information by department, classification, data-type, or access-type is common. Each organization or sub-organization may have different policies to enforce. This type of segregation, in many cases, requires redundant systems. For instance, each classification level may have a different file server, print server, e-mail server, and web server.

A computing system that allows individuals access to information that carries classification markings of different levels is called a "Multilevel" system. Computing products that undergo the rigorous independent evaluation testing required for high assurance systems to ensure that information is handled in accordance with established organizational policy carry the name "Multilevel Secure" (MLS). [Ref. 2] These products undergo independent evaluation under the Trusted Product Evaluation Program (TPEP) and many have been rated against the criteria of the Trusted Computer System Evaluation Criteria (TCSEC or "Orange Book"). [Ref. 3] A newer evaluation program relies upon what is colloquially called the "Common Criteria" (CC) and is currently in use and governed by ISO/IEC-15408. [Ref. 4]

Current hardware and software products capable of providing the type of assurance that Department of Defense (DoD) and other government organizations often require have many drawbacks. The most noted drawback to existing multilevel secure devices is the interface provided to the user. It is archaic, primitive, and difficult to work with. These systems do not allow access to the popular office automation software that is prevalent in industry. On the job, those working in secure environments must use these systems, but at

home, they use Commercial-Off-The-Shelf (COTS) software and other sophisticated information technologies to write personal letters, view e-mail, and create web pages. These users may ask, "Why can't I use this type of software at work?"

There are several approaches to providing access to information to personnel in a multilevel secure context. One possibility is to place a physically separate single-class computer at the desk for each classification the user needs to access. This would provide an individual access to the material they need while using COTS hardware and software, but this does not provide access to multiple levels in a single connection. Access policy may allow an individual with a certain classification to view information at a security level and all levels that are dominated by that level. A single class connection is referred to as "system high." The system contains no mechanism to reliably label information. This means that all information, no matter what the original classification, must be classified at the "system high" level. This is not true multilevel access, as the original classification of the information is not preserved. Indeed the data would need to be duplicated on some other device with appropriate classification to maintain the original data at its original classification. This solution is not acceptable in environments where classification levels are encountered. It requires the purchase of too many COTS PCs and software licenses because one unit would be required at each desktop for each classification level supported. This also allows the user to store documents without the labels necessary to ensure that the document is handled properly if converted to hardcopy. Another possibility would be to place a high assurance multilevel device at each desktop. This would allow the user to access multiple levels of information as allowed by the local access control policy with sufficient assurance that the data would remain segregated. Another benefit is that this can be done with a single unit at the desktop. Unfortunately, this solution has several drawbacks. Foremost among these is the fact that these devices are expensive. In addition, they provide an archaic human-computer interface and do not allow the use of COTS software that provides the office productivity tools users desire. Ultimately both of these alternatives are unattractive because they do not provide full access to material in accordance with organizational policy and the use of COTS hardware and software.

The first solution is commonly used in the Navy and other DoD services. This is driven by the need to supply the user with the office productivity software that she/he desires. The visionary documents for the future of the Navy and eventually the entire defense infrastructure [Ref. 5] [Ref. 6] utilize a similar structure where COTS PCs using COTS software access material from discrete system-high domains. For this reason we must seek a better solution that provides us the full access that policy allows, uses COTS hardware and software, and is more affordable. The Naval Postgraduate School Multilevel Secure Local Area Network project (MLS LAN) sets as a goal to provide access to multiple levels of information in accordance to some mandatory security policy while maintaining the integrity of the label attributes originally associated with the material. While providing this high assurance access, we give the user access on COTS hardware running COTS software that meets the high demand for a well-developed human-computer interface.

B. GOALS/THESIS QUESTIONS

The MLS LAN project utilizes a client server solution that provides multilevel access to information through high assurance servers by client COTS PCs running COTS software. This combination allows access to affordable hardware and software with state of the art human-computer interfaces thus reducing cost while maximizing productivity. This also facilitates access to shared resources in such a way that communicating, collating, and disseminating information is much more efficient. The solution must provide this access to shared resources with assurance that the information is not leaked from high to low classification levels. COTS hardware and software is accessing this information thus providing the user-friendly office productivity software on PCs that are affordable or better yet, already owned. So steps must be taken to ensure that attackers who gain access to the system do not cause the unauthorized exposure of sensitive information.

The server element of this solution is carried out by a modified Wang XTS-300. It is used as the locus of policy enforcement making the component added to the client workstation a support element that is not required for policy enforcement. The Wang XTS-300 is a high

assurance multilevel secure computing device that has been rated under TCSEC to Class B3. This device will provide the mandatory and discretionary access controls to information and user identification and authorization (I&A).

A Class B3 rating requires the use of security kernel technology within a well defined trusted computing base (TCB) to enforce policy. Elements defined in the "Orange Book" for policy in a high-assurance computing device include: Discretionary Access Control, Object Reuse, Labels for devices and subjects/objects, Mandatory Access Control, Identification and Authentication (including the use of a Trusted Path), Audit, Operational Assurance (including system architecture, system integrity, covert channel analysis, trusted facility management, and trusted recovery), Life-Cycle Assurance (including security testing), Design Specification and Verification, and Configuration Management. All these considerations are absolutely necessary to the successful evaluation and certification at Class B3.

This paper attempts to define the requirements for connecting a network client personal computer to this high-assurance computing device to create a distributed TCB able to enforce mandatory policy with the assurance of the original high assurance device. The mechanism used to connect the client to the server must at a minimum:

- Establish a session under the control of the server, and .
- Ensure that residual information that may have been stored on the client does not result in unauthorized information flow.

Specific questions to be addressed in this thesis are:

- What must be done at the client to support the policy enforced by the server TCB?

Client PCs in common network environments are insecure devices. In addition, there are no integrity checks before the operating system is loaded. The operating system provides only weak assurance of policy enforcement, and the applications running on the operating system are untrusted. These factors all combine to make the client an ideal entry point for an attack in

a network. This thesis seeks to describe a means of providing support for identification and authentication of the user and hardware to the TCB, supporting trusted path services, improving assurance of a known initial state of the PC by properly bootstrapping the PC, loading the operating system and applications from a controlled read-only source, and cleansing the PC between sessions.

- Where in the client is the assurance element going to reside?

The TCBE is an add-in card for a PC. This question seeks to determine where in the PC this card should be added. The need to address this question rests in the necessity to place the card in a position where it can maintain sufficient control of the host PC while maximizing its ability to communicate with the host processor, applications, and the rest of the network. One requirement that must be maintained is that the trusted path is always reachable by the user no matter what operations are taking place in the PC.

- What controls are required in the client workstation for the hardware?

Security begins from the ground up. The need to have a solid known foundation to start from is paramount. Once the system is running, the operating system is loaded, and the applications are accessing information, what must happen to re-establish communications with the High Assurance Server (HAS) trusted path? (A Trusted Path is an isolated, secure, unspoofable communications channel between the user of a secure system and the policy enforcing trusted computing base of that system. It is used to access functions that require a high degree of trust to accomplish what they are designed to do such as login and change session level.) How are communications other than those of the trusted path stopped? What must be done to the operating system while these communications are taking place? If a new session is negotiated, what must happen in the PC? All these issues cover the security of information temporarily

accessed on the PC and the integrity of the hardware on which the operating system and applications are run.

- What controls are required in the client workstation for the operating system?

During times when the trusted path is invoked all other communication from the PC to the rest of the world must be stopped. Only the trusted path communications are allowed. This is a confidentiality and integrity concern. Confidentiality of the data temporarily accessed on the PC must be maintained to insure that no information is leaked. This must be analyzed because many commercial operating systems cache information for efficiency. This information must be inaccessible to unauthorized subjects. Integrity of the trusted path communications must be maintained to provide the necessary assurance that the user is in communication with the TCB and not some other malicious entity.

To answer these questions, a careful analysis of the current technology in PC systems was conducted. Conclusions regarding the client PC hardware will be discussed. The proof of concept prototype is based on an Intel Pentium II motherboard using the 440BX chipset with 128 MB of RAM, PCI based NIC, sound card, IDE hard drive and controller, and CD ROM. Some specific information about this configuration is located in Appendix A. The prototype is running the Windows NT Workstation 4.0 operating system with Service Pack 5 installed.

A representative model of the prototype LAN is shown in Figure 1. In this model, the High Assurance Server (HAS) provides the access control elements and enforces the security policy. The application protocols run on the HAS. These provide the services and access to shared resources that users need. The extension of the trusted computing base to the client PC provides a means of controlling the client PC so that its access to information will be controlled by the policy being enforced at the high assurance server. The TCBE will support a trusted communications channel to the server so that server-based trusted path interfaces can be exported to the PC with high confidence. This communications channel will permit the PC hardware and the user to verify themselves to the HAS and the HAS to the user. It will

support server communications over an untrusted LAN during user sessions ensuring their privacy and integrity using cryptography. A major focus of this thesis is to discuss the features necessary within the Trusted Computing Base Extension to extend the system security perimeter to the client PC.

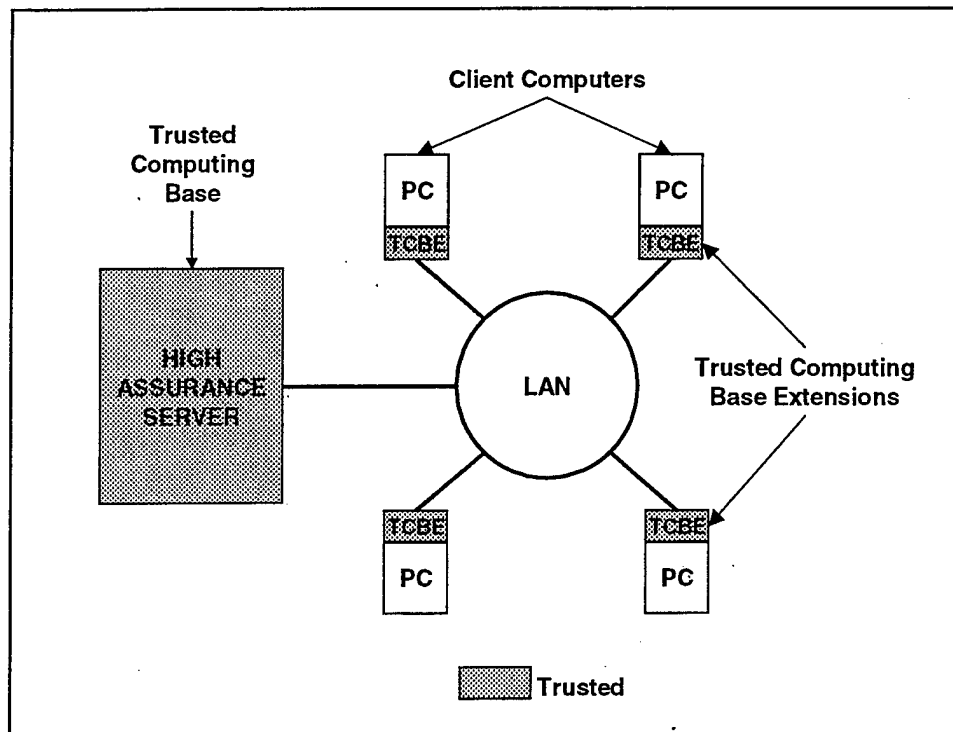


Figure 1. A Basic MLS LAN Architecture

Figure 1 represents the basic MLS LAN architecture. The high assurance server (HAS) enforces the security policy and controls access to information. It provides a protocol that is used for client access to information. The TCBE is the trusted component within the client PC that provides support for the policy dictated by the HAS trusted computing base, a means of establishing communications with the HAS, and the necessary mechanisms to ensure secrecy and integrity of all communications.

1. Trusted Computing Base Extension (TCBE)

a. Why do we need a TCBE?

The Trusted Computing Base Extension (TCBE) is required because COTS PCs running COTS operating systems and COTS applications are not trusted or trustworthy. Attacks on personal computer hardware and software are on the rise. Attacks on BIOS, such as the Chernobyl virus, have rendered machines and their data unreachable by overwriting crucial sections of the BIOS. Other attacks in the form of viruses, such as Melissa, use personal data to propagate themselves across networks by imbedding macros that run programs such as Microsoft Outlook, without the user being aware that they are working. Commercial systems offer little in the form of security. Discretionary Access Control (DAC) presents no protection against the threat of Trojan Horses.

Chipsets used to run the hardware on PC motherboards are constantly being updated. BIOS flash-ROM, PCI chipsets and bridges, and Accelerated Graphics Port (AGP) chipsets exist on nearly every motherboard to assist the CPU in efficient operation. These are very powerful and versatile computing devices in themselves and present open space to sequester information during sessions at a high access control level. This information may be exfiltrated later by malicious code when the user is logged on at a lower access control level. Both hardware and software, without modification, leave open vulnerabilities to attack. [Ref. 7]

System security policy is enforced by a TCB. The TCB must be able to establish trust relationships with external connections, otherwise its policy enforcing functions could be completely short-circuited. Thus, in computer security engineering, it is important to describe how the policy is enforced when components are interconnected. Ultimately, policy enforcing software, hardware, and firmware should be subjected to independent evaluation of its ability to enforce the properties of the reference monitor. [Ref. 8] A careful architectural approach permits individually evaluated components to be combined in a secure networked

system. The TCSEC and the Common Criteria have set procedures for connecting trusted components together in network environments in such a way that the whole can be evaluated.

The features that make up a TCBE cannot be achieved through arrangement of COTS hardware and software alone. These features must be implemented in specialized software, hardware, and/or firmware that can be evaluated to carry out their designated tasks and fully support the TCB in enforcing organizational security policy.

Not all organizations require this type of security. High assurance security is required in DoD installations to ensure that highly sensitive information that could cause grave damage to national security is not lost or compromised. To date, industry has protected its information assets using other mechanisms. However, high assurance secure solutions provide enabling technologies for commercial use of the Internet.

b. What is a TCBE?

A Trusted Computing Base Extension (TCBE) is just that, an extension to an existing Trusted Computing Base (TCB). A TCB is "the totality of protection mechanisms within a computer system -- including hardware, firmware, and software --the combination of which is responsible for enforcing a security policy." [Ref. 2]. The Common Criteria calls these mechanisms in a Target of Evaluation (TOE) a set of Trusted Security Functions (TSF). The trusted security functions are relied upon to correctly enforce the TOE's Security Policy (TSP). [Ref. 9]

A TCB Extension (TCBE) logically extends policy-supporting mechanisms of a TCB to some remote device. As an extension to the TCB, it must do nothing to weaken the TCB's abilities to enforce policy. It supports the TCB's enforcement of policy, not necessarily enforcing policy in itself. By nature of the fact that the TCBE is remote from the TCB, it must have a communications path with the TCB that can be trusted. These communications functions will support a trusted path between the high assurance base and the user. This path has certain required characteristics:

The TCB shall support a trusted communication path between itself and users for use when a positive TCB-to-user connection is required (e.g., login, change

subject sensitivity level). Communications via this trusted path shall be activated exclusively by a user or the TCB and shall be logically and unmistakably distinguishable from other paths. [Ref. 3, p.107]

The Common Criteria refers to the trusted path as a means of communications between a user and a TSF with necessary confidence to support the TSP. The communications path between two separate trusted components (IT products) is referred to as a Trusted channel. The trusted channel must support the TSP with assurance. This gives rise to the first and most important function of the TCBE: to facilitate creation of an unspoofable, secure, and private pathway of communication between the remote device and the TCB.

Throughout this thesis some common terms will be used. They are:

- Host – A personal computer running COTS software connected as a client in a MLS LAN.
- Server – In the MLS LAN the server is the high assurance server that provides the locus of policy enforcement and the access to shared resources through applications protocols.
- TCBE – A trusted mechanism added to the host to support the security policy of the server.

There are several capabilities that must be included in components that establish trusted channels of communications between trusted elements in a network environment. These capabilities, and features that the TCBE must have to support TCB policy enforcement at the client, are enumerated below. They will be discussed in detail separately.

- Secure Communication – over a network connection require encryption and the protocols to support encryption
- Unspoofable – require absolute control over the signaling device to establish communications with the high assurance elements of the system. This means that the TCBE will have to have keyboard and monitor capabilities. These will be either separate from the host or under the TCBE's absolute control.

- Supports TCB in enforcement of policy. This requires absolute control over the host device to:
 - Prevent the leakage of information from high to low. It must control the host computer's access to I/O devices to ensure that the reference validation mechanism at the server is invoked to mediate access to information.
 - Control access to information such that it is only accessible to those with sufficient authorization. It must address object reuse issues in the PC host computer, operating systems that run on the host, and applications that run on the operating systems.

The PC in which the TCBE will reside, is insecure, so it must be made as secure as possible. The TCBE can be made to assist with this process. First, the TCBE will need to create a firm foundation for all other operations. This can be accomplished by carrying out steps similar to the AEGIS architecture for secure and reliable bootstrap. [Ref. 10] In this architecture, the Basic Input Output System (BIOS) from Power On Self Test (POST) to operating system bootstrap is controlled and the integrity of each step in the process is verified before it is allowed to execute. A framework is created to reload sections that fail integrity checks to create a system that is more reliable.

The TCBE must address secrecy issues on the PC to prevent the leakage of information from "HIGH" to "LOW". The TCBE can be made to do this by controlling all of the storage locations available on the PC. This is principally an object reuse issue. These locations include, but are not limited to: RAM (system and graphics), FLASH-ROM (BIOS firmware), registers, buffers, bridges, cache memory, and permanent media (hard drive). Many of these potential storage channels are defined in the BIOS vector tables, as they are required to be accessible. These devices can be "cleansed" or purged by cycling power to the PC. Permanent media is most important to a PC's operation. It retains the operating system, applications, and information that the user may wish to store there. Operating systems and applications today – especially Windows products – are highly configurable. This feature

carries with it a requirement that they be run from a media that can be written to. In fact, these operating systems often modify themselves on startup, recording platform specific information such as devices attached to the PC and user specific information such as profile information. Operating systems and applications also cache information the movement of which might violate organizational policy. This information could be moved by malicious code from HIGH to LOW. Such a potential leakage must be prevented. This requires more stringent enforcement of object reuse at the client than commercial operating systems are able to accomplish. A strong enforcement of object reuse issues at the client workstation is one of the principle problems that this thesis must address.

C. ORGANIZATION OF THESIS

Chapter I provides a detailed statement of the problem. Chapter II breaks down the TCBE into functions and discusses the necessity of each function to carry out its designed tasks. Chapter III describes the disk-based methods of delivering the operating system to the client and the security ramifications of each method. Chapter IV discusses the use of Windows NT Server, Terminal Server Edition, with diskless network PCs and thin clients. It analyses many possible topologies and assesses the security ramifications of each. Finally, in Chapter V, the findings are summarized and suggestions for furthering the design and implementation process are given.

Appendix A presents some basic information about modern PC architecture, the Basic Input Output System (BIOS), and a particular attack on BIOS. Appendix B gives the features of Norton Ghost by Symantec, a commercial program for rapidly copying hard-drives and partitions.

THIS PAGE INTENTIONALLY LEFT BLANK.

II. FUNCTIONAL DESIGN REQUIREMENTS FOR A TRUSTED COMPUTING BASE EXTENSION

This chapter attempts to expand on the work conducted by J. Hackerson in [Ref. 11]. In it, he enumerates five functional requirements for the Trusted Computing Base Extension (TCBE). They are: (1) prevent object reuse and data remanence, (2) the high assurance server (HAS) controls all read and write access to information by users, (3) the TCBE must support the trusted path (between the HAS and the TCBE), (4) the TCBE will be modular, concise, and protected against tampering/interference, and (5) the TCBE will mediate all access to the HAS. All these requirements derive from the need for security and the support of the XTS-300's TCB. This chapter seeks to define the physical interfaces that stem from its location on the host PC, its interface with the user for trusted path operations, and its interface with the rest of the LAN. The first section is an identification and analysis of probable interfaces. The interfaces will be identified, their purpose will be given, examples of how they might be implemented/controlled are explained, and finally, the security aspects/ramifications of the potential implementation/control path will be discussed. Finally, a summary and conclusion will be made from the analysis conducted in this chapter. First, the purpose and general function of the TCBE is restated.

A. PURPOSE OF THE TCBE

The Trusted Computing Base Extension (TCBE) resides in the client personal computer and connects the client to the high assurance protocol server (HAS) via Ethernet connections. It will give the user at the client access to shared resources controlled by the HAS. The HAS is capable of maintaining the separation of multiple levels of classified information and grants access to that information in accordance with an organization's access control policy. Typical access control policies allow access by individuals who have clearances that dominate the classification of the information to be accessed. The TCBE must provide mechanisms to support the high assurance server's access control policy. These mechanisms

require identification and authentication (I&A) mechanisms that are located within the HAS. These mechanisms rely on the principle that the user and the trusted computing base are in direct communications with each other and not some hostile code. This communications mechanism is called the trusted path. Its characteristics will be described shortly.

Accessing information in accordance with policy may allow the user to read multiple levels of information concurrently up to some user-declared level (where the user-declared level is less than or equal to the maximum he/she is allowed by the organization's access control policy – and indeed this is the goal!) This information will be passed to the user at the client workstation. The client is not capable of ensuring that there is no information stored on itself when there is a change in access level (especially to a lower access classification level.) To provide assurance that information is not stored and subsequently accessed by unauthorized individuals, proper object reuse must be carried out on the client.

The TCBE must be able to function securely in a hostile environment. There must be assurance that the TCBE is doing what it is designed to do. The TCBE will obtain power from the host computer and must present an interface, through one of the PC's expansion buses, to operating systems and applications running on the host. Services will be provided to the host's operating system and applications through drivers. Because of this physical and software interface, the TCBE must be able to protect itself against attacks that could cause unauthorized information flow.

1. Trusted Path Communications

As with any security-related product, the TCBE must provide elements of security in a possibly hostile environment. Information assurance is the sum of confidentiality, integrity, and availability. In the context of a LAN, the TCBE must securely extend the TCB of the XTS-300 to the client PC. To do this, the TCBE must have a trusted communications path to the TCB. For the MLS LAN, this is intended to occur over an Ethernet LAN connection under TCP/IP protocols. The work has already been done at the XTS-300 to accept connections of this type, and a minimal specification has been made for creating a trusted path connection between the XTS-300 TCB and an external trusted component, the TCBE. [Ref. 12]

The TCBE must be designed to complete this interface, supporting all security requirements of a trusted path connection. TSF stands for "TOE Security Functions". "TOE" is the "Target of Evaluation" or the information technology product that is being or has been evaluated under the Common Criteria. The TSF are:

"A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the TSP." [Ref. 9, p. 6]

The Common Criteria describes trusted channels and paths as having the following general characteristics:

- The communications path is constructed using internal and external communications channels (as appropriate for the component) that isolate an identified subset of TSF data and commands from the remainder of the TSF and user data.

- Use of the communications path may be initiated by the user and/or the TSF (as appropriate for the component)

- The communications path is capable of providing assurance that the user is communicating with the correct TSF, and that the TSF is communicating with the correct user (as appropriate for the component) [Ref. 13, p. 163]

The Common Criteria divides its guidelines into Functional Class, Families, and Components. Each Functional Class describes the functions the family will support. It contains the requirements for that family. Below "FTP" identifies the class. It stands for "Trusted Path/Channel". "TTC" is the family name and stands for "Inter-TSF Trusted Channel". For communications between trusted information technology products (i.e., XTS-300 and TCBE):

FTP_ITC.1.1 The TSF shall provide a communication channel between itself and a remote trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2 The TSF shall permit [selection: *the TSF, the remote trusted IT product*] to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for [assignment: *list of functions for which a trusted channel is required*]. [Ref. 13, p. 164]

The TSPs are the TOE security policies. TSF are therefore essentially the same as the Trusted Computing Base (TCB) defined as:

“The totality of protection mechanisms within a computer system, including hardware, firmware, and software, the combination of which is responsible for enforcing a security policy. A TCB consists of one or more components that together enforce a unified security policy over a product or system. The ability of a TCB to enforce correctly a unified security policy depends solely on the mechanisms within the TCB and on the correct input by system administrative personnel of parameters (e.g., a user's clearance level) related to the security policy.” [Ref. 2]

From the above characteristics, the TCBE must be able to assist in creating a secure communications channel that is logically separate from other network communications. Trusted path communications must also provide for the high-integrity identification of both sides of the path for both the hardware and the user. The XTS-300 TCB already provides a Trusted Path to serial ports. The TCBE must participate in a new protocol with the XTS-300 that will provide for a trusted path. This protocol will contain mechanisms to ensure the confidentiality and integrity of the trusted path. Obviously, a TCBE-to-XTS-300 trusted path could be constructed using a separate communications line (i.e. a physically separate wire between the entities.) This is not the desired solution. Instead a logically protected trusted path will be constructed. The TCBE must have mechanisms for protecting trusted path communications from disclosure and modification. It will also have mechanisms for initiating the trusted path.

The TCBE must be designed to prevent any communications out of the device by the host operating system or any applications requesting services of the TCBE during trusted path communications. For this purpose, the TCBE must be in direct control of the NIC interface between itself and the operating system/applications.

2. Object Reuse

As stated earlier in this thesis, object reuse is the re-issuing of storage objects such as memory, disk space, buffers, to a subject after a previous subject has returned the object to a general pool. There are two general methods for ensuring proper object reuse. One method

overwrites the storage object before issuing it to the new subject. Using this method the object could be overwritten when the storage object is returned to the pool. The other method withholds read access to the object from the new subject until that subject actually writes to the storage object. In the context of the NPS MLS LAN, the subject is the Windows NT client workstation and includes: operating system, memory, buffers, hard disk, registers, cache, etc.

The PC contains a large variety of storage objects in the form of registers, and buffers for use by peripherals, operating systems, and applications to perform various functions. The TCBE must have control over these storage objects to prevent unauthorized information flow when a session level has been renegotiated. The TCBE must also have sufficient control over the host PC to be able to clear all its registers, buffers, and other volatile and non-volatile memory spaces.

B. TCBE EXTERNAL INTERFACES

The TCBE, in order to carry out its primary duties, must be able to interface with the user and the host computer. This interface comes in the form of input from the user and display to the user. Input could be through keyboard, smart card reader, an electronic switch, or a combination of all these methods. The output could be through a fully feature graphical user interface or just a liquid crystal digital (LCD) display. The TCBE must also interface with the PC in such a way that it has extraordinary control over the PC itself primarily for control of object reuse. One of the elements that is part of any PC is the permanent media, such as the hard drive. The hard drive stores the operating system and applications, as well as information that user wishes to access repeatedly. Object reuse control of hard drive storage objects is one of the most difficult duties of the TCBE. The TCBE must also interface with the XTS-300. As already stated, this interface is designated to be over an Ethernet connection using a TCP/IP protocol. The functionality of this interface is seen today in the common network interface card (NIC). Control over the PC for the purpose of object reuse and the control of information flow to input/output devices requires that the TCBE occupy one of its two expansion buses ISA/EISA and PCI. ISA stands for "Industrial Standard Architecture" a standard that defines

the bus structure, architecture, and clock frequency. EISA stands for "Extended ISA" and is the standard a bus fully compatible with ISA but with a 32-bit data bus width. PCI stands for "Peripheral Component Interconnect" and is another bus standard that has a higher clock frequency, and the capability to allow data transfers without direct supervision of the CPU.

Each of these TCBE interfaces is discussed separately. Why we need the interface, how the interface may be implemented, and what security benefits/risks arise for possible implementations are addressed.

1. Background

There is a common method of dealing with hardware accessible to a computing device. When the hardware needs the attention of the CPU, it issues an interrupt. Interrupts used by peripherals are called Interrupt Request Lines (IRQs). These are maskable interrupts, which means that the CPU can ignore them if something more important is going on. IRQs are prioritized. This means that they will be handled as if placed in a prioritized queue. Those of higher priority will be serviced first. Below is a table of interrupts in order of priority.

IRQ Number	Device Assigned
0	System Timer
1	Keyboard
2	Slave Interrupt Controller
8	Real-Time Clock
9	Redirected to IRQ 2
10	
11	
12	PS/2 Mouse
13	Math Coprocessor
14	Hard Disk Controller/Primary IDE
15	Secondary IDE
3	COM 2/ COM 4
4	COM 1/COM 3
5	LPT 2
6	Floppy Controller
7	LPT 1

Table 1. Table of IRQ Lines Assigned [From Ref.14, p. 53]

Hardware interrupts such as the IRQs above are translated to software interrupts. Software interrupts are access points into BIOS code. When an interrupt request line is activated and recognized by the CPU, the CPU saves its state and transfers control to the software that services the interrupt. This software is stored in the BIOS. Operating systems can “hook” into these sections of code to communicate with the hardware devices rather than run their own code to manipulate the device. Many devices, however, are accessed directly by the operating system rather than through the BIOS as the operating system code is more efficient. This is often due to advances in motherboard controller chipsets occurring more rapidly than updates

to BIOS code. The disk drive unit is one such example of a device whose BIOS routines are bypassed.

PCI devices may be assigned an "INT" number, which is the equivalent of an IRQ but controlled by the PCI bus controller. PCI INT numbers may be used to associate an IRQ with a particular PCI card. Each PCI slot (receptacle for an add-in card) is able to activate (or own) up to four PCI INT numbers. These assignments of PCI INT and their subsequent assignment to an IRQ can be controlled somewhat. The method of re-mapping PCI INT to IRQ is called "steering" and can actually allow more than one PCI device to share the same IRQ line. [Ref. 14]

2. User Input to the TCBE

a. Why a User Input?

User input is absolutely required for Identification and Authentication (I&A) processes with the High Assurance Server (HAS). The current method used by the XTS-300 utilizes a user ID and a password to determine the identification of the user and to subsequently grant accesses according to the policy established on the XTS-300. The user must be able to key-in information.

The user must be able to initiate a communication path. This will be the trusted path (TP) with the Trusted Computing Base (TCB). The method of signaling the TCB so that such a path may be constructed is often referred to as use of a Secure Attention Key (SAK). This key must generate a secure and direct signal to the trusted path function, it must be unique, and untrusted software must not be able to generate or mimic it.

b. How To Get User Input

PC keyboards are miniature computers in themselves. A scan matrix is made up of crossed conducting lines. At each crossing is a pressure switch under a key. Depressing a key closes a switch that is interpreted by the keyboard processor to generate a scan code. The scan code is captured, sent to a buffer, and transmitted to the motherboard. Depending on

the operating system running on the PC, a keyboard program or a keyboard driver interprets the scan code into alphanumeric values according to the language(s) supported. In Windows NT, the keyboard driver is loaded automatically by the system each time the system is started. Most PC motherboard manufacturers include a keyboard interface on the motherboard. All operating systems and applications are programmed to access this well-known device. Access is controlled by Basic Input/Output System (BIOS) interrupt service call to let the CPU know that there is input from the keyboard.

One way to get the keyboard input to the TCBE is to create routines for the BIOS to route all keyboard input through the TCBE so that scan codes that generate the SAK can be interpreted and acted upon. To do this, the BIOS mechanisms that route the interrupt generated by the keyboard to the service routine must be changed. This requires modification of the main motherboard BIOS. It is not the most desired solution as will be seen in the security analysis that follows.

Another method of acquiring keyboard input by the TCBE would be to have a dedicated keyboard input device controlled by the TCBE. This device would be a standard keyboard. It will have direct input into the TCBE. This will require that the TCBE act the same way that the PC does with respect to handling keyboard input. It must have a way of detecting when the keyboard is pressed (a BIOS interrupt) and then the software to interpret the scan code as a key being pressed. This method does require additional hardware on the TCBE in the form of a connector, possibly buffers, and a data path to the CPU on the TCBE for processing the scan codes.

Other methods of gathering user input could be devised, but they will be left for future work. The easiest and most well known form of user input to any device is the keyboard.

c. Security Analysis of Keyboard Sources and Control

Keyboard input from the motherboard carries with it reliance on elements outside of the TCBE that are too easily subverted. There is a no trust in the keyboard driver loaded by the operating system to properly interpret and re-vector the keyboard information to

the TCBE for proper action. These drivers are too easily accessible to the knowledgeable user, and present an attack route that is too easily implemented. A simple attack would cause the keyboard driver to invoke a program that displays identical text strings that the XTS-300 would generate if the real trusted path were initiated by pressing the SAK. In this way, the attacker could first gather the user ID and password pair, act as if there were an error in the input, and then forward the SAK sequence to the TCBE to wake-up the real trusted path. This violates the integrity of the trusted path operations and the identification and authentication process. The TCBE must be able to demonstrate the ability to establish a trusted path with the TCB on the XTS-300. Using untrusted code to interpret the keystrokes made to generate the SAK is not acceptable.

Keyboard input from the TCBE presents a much better solution. The connection would be dedicated to communication from the user directly to the TCBE. This eliminates the need for trust in the host PC operating system to properly handling the keystrokes and the difficult task of modifying the system BIOS to forcibly re-vector the keystrokes to the TCBE. The amount of code it takes to get this input is small and therefore verifiable. The amount of trust that the user and the TCB can have that the input is coming directly from the user is amplified. From a security standpoint, this solution provides the highest degree of integrity and assurance. These benefits outweigh the presence of an additional component attached to the TCBE and an additional keyboard on the user's desk.

3. TCBE Video Display to the User

a. Why a Display to the User?

Display to the user is necessary for the same reasons as the user input to the TCBE. A display must be trusted to present the correct information from the High Assurance Server for all Trusted Path communications with all the integrity and security of the trusted path. When the user communicates via the trusted path, there must be assurance that the displayed text is coming from the TCB. When the user invokes the Trusted Path, a response is expected. The current method of communicating to the TCB of the XTS-300 is text based.

For instance, initial communications to a user presents a logon prompt for user ID and then a password. When this is complete, a default session level and integrity level is assigned or alternatively, a prompt for session level and integrity level is presented so that the user may choose the session level within a predefined range. Each time the SAK is pressed, the trusted path connection must be invoked permitting the user to access trusted path functions. Each of the foreseeable functions that will be accessible via the TCBE will require visual display to the user. In fact, providing visual display to the user is one of the principle reasons for a trusted path. Lack of such a display can result in serious security problems. [Ref. 15] This requires assurance that the user display is receiving information from the XTS-300 within the trusted path. For these services a low bandwidth display is sufficient.

Once trusted path communications are complete and the PC begins applications, the information transfer becomes much more varied. Instead of text-based interplay between the user and the resource provider, there might be full motion video, multi-bit color graphical images, or web content with all its graphical capabilities. These communications do not require the same assurance as the trusted path. For this type of communication, a high bandwidth display is needed. The prototype system is planned to provide an IMAP e-mail service. The services will be accessed through the commercial software provided by e-mail browsers such as Eudora, Outlook, and Netscape. These browsers are graphical in nature. For these services a high bandwidth display is needed.

b. How to Obtain Display Functionality for TCBE Functions.

Modern PCs require display units. The current state of the art in display adapters is the Accelerated Graphics Port (AGP). These devices are designed to access system memory shared with the CPU without using the PCI controller directly. This creates a high bandwidth display processor capable of moving millions of bits of graphical information at speeds approaching 533 Mbytes/second.

The figure below shows typical motherboard architecture with the AGP/PCI chipset. Normal access to the display unit is via the Host Adapter by code executing on the CPU. There are no documented ways to access the AGP adapter directly from another source

such as an adapter card on the PCI bus. Accessing the AGP adapter from the TCBE could be accomplished by exporting a program to be executed on the main CPU. Once the program is executing on the main CPU, a data stream will be established from the TCBE to the program in execution to pass the text objects for display. This includes information coming from the TCB via the Ethernet connection and echoed keystrokes entered by the user on the local keyboard. This requires transporting code and raw textual information from a trusted device to an untrusted device with no verifiable means of ensuring that the code is executed without modification.

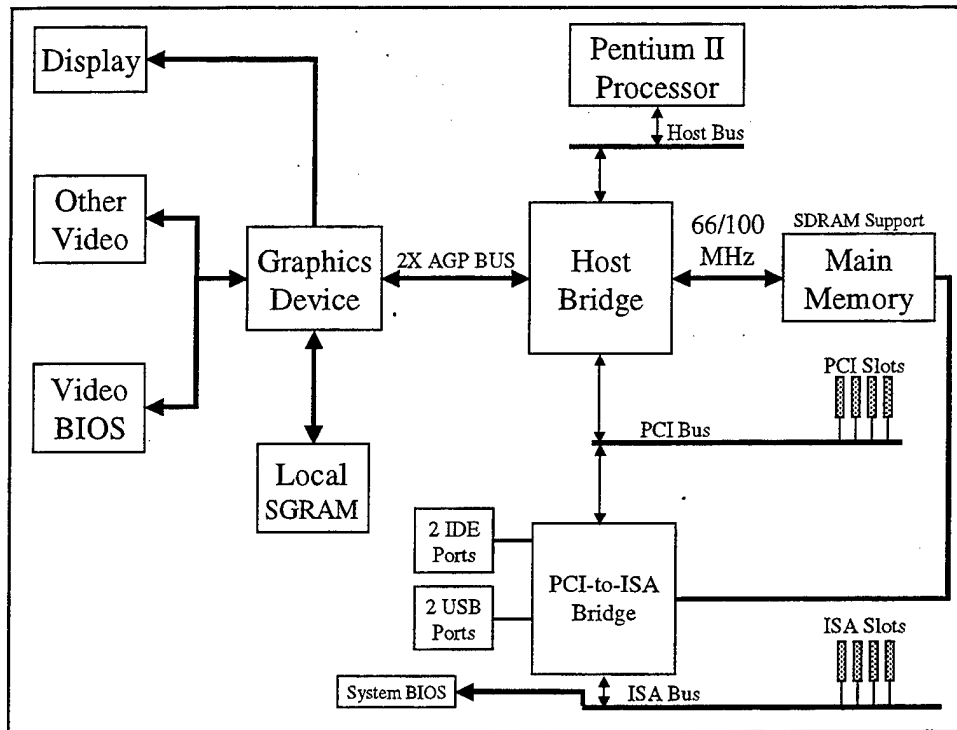


Figure 2. Typical Intel Based Motherboard Showing Graphical Units

Figure 2 is representative of a modern motherboard. It shows support for the advanced graphics port (AGP), IDE controllers and USB ports on the motherboard. It shows the relationships between the CPU, host bridge PCI bus and the ISA bus. Communications and data paths are represented by the double-ended arrows.

Another way to acquire display that has sufficient capability and a high degree of assurance is to implement one on the TCBE. The display could be a simple Liquid Crystal

Display unit capable of displaying one or two lines of text like one used on a graphical calculator. These units are unobtrusive to the user and could even be attached to the corner of a typical keyboard unit. Minimal hardware additions are need on the TCBE. A graphics controller chip is needed to send signals to the display. A simple DB-15 or DB-9 connector would be sufficient for moving the signals to the display from the card. The unit would display monochrome text only, so there are essentially no additional requirements for memory outside a moderate buffer used to store the text. Closely coupled with the graphics controller is a character generator that actually shapes the pixels to form the characters to be displayed. All of this would fit nicely into a single silicon package that takes a minimal amount of room on the TCBE card.

c. Security Analysis of Display Sources.

The use of the native display capabilities of the main motherboard is ill advised. The only conceivable way of utilizing that display is through the main CPU, which requires exporting code from the TCBE to the host bus for execution in a potentially hostile environment. This is impossible to accept for trusted path operations because it breaks the chain of verifiability to the user for high assurance communications with the TCB.

The improved assurance factors and ease of implementation compared to the use of the main CPU make implementing the display as part of the TCBE a better choice. As with the addition of the keyboard to the TCBE's functionality, this addition requires minimal code addition and maintains verifiability and conciseness of trusted code. It also offers a sure protection against spoofing a trusted path connection and causing loss of user account data through such an attack.

4. TCBE Interface with the Personal Computer

a. Why a Personal Computer Interface?

To add features to a personal computer, one must design and construct a motherboard that has all the desired features, or design and construct an add-in board that can

occupy one of the two available expansion buses existing on modern motherboards today. The former is impractical and the latter presents one acceptable choice.

The reasons for using an expansion bus are many. First, the PC bus system was designed to allow developers to create feature-rich devices for use by the general public through standardized connections offered in such a bus, and supplying the device developers with an interface chipset capable of communicating with the main motherboard CPU and the system memory. Second, the bus was designed to allow decoupling of the expansion card capabilities from the CPU. This means that devices with independent computing power, could function without needing to access the CPU. It also means that devices in separate slots could exchange information without accessing the CPU. Each of these characteristics brings great versatility and risk. Third, the development of the bus structure allowed adding many devices and therefore many functions to the PC without a requirement for additional power connectors. This is possible because the standards that define the buses also define limits of voltage and current drawn by the add-in cards.

In our design, the TCBE will exist on an expansion card. It will draw its power from the motherboard through the expansion slot, control the PC through the available communications lines, and provide information to the host operating system and applications. It requires a large bandwidth to potentially supply information delivered on a high-bandwidth Ethernet connection. It will be non-bypassable in that all PC communications to the LAN must pass through the TCBE. In addition, the TCBE must be resistant to tampering by entities executing on the host.

b. Choices in the Personal Computer Interface.

There are two expansion buses supported by motherboard manufacturers today. The Industrial Standard Architecture (ISA) bus is kept for legacy cards. There is currently little or no active development for the ISA bus. The Peripheral Component Interconnect (PCI) local bus, on the other hand, has considerable developmental activity in industry.

The PCI standard is ever expanding to support greater functionality and bandwidth required for devices that are being developed today. The types of devices being developed for the PCI bus that support the initiative of the NPS MLS LAN project include: high transfer rate hard drive controllers, gigabit Ethernet network interface cards (NICs), and high performance cryptographic coprocessors that have general programming capabilities. [Ref. 16]

Developing expansion cards for the PCI bus requires an in depth understanding of the PCI standard and Expansion BIOS capabilities. It also requires sophisticated hardware to assist in developing and troubleshooting devices. With the hardware and software development tools in hand, it will be possible to design and test the TCBE for use on the PCI bus module by module. Each module must consist of well-defined interfaces so that module dependencies can be well defined. Each module can be developed, implemented, and tested before adding another. In this way, the capabilities of the TCBE can be increased incrementally.

It might be possible to develop an add-in card that can control other add-in cards and thus shorten the implementation of a prototype unit. This may be accomplished by managing the PCI slot configuration. A PCI card can be assigned a PCI INT number (usually assigned by the slot number i.e., slot 1 is INT #1), and that PCI INT number can be mapped to the IRQ line. IRQ lines are used for signals between the CPU and a device that provides services for the CPU. For example, when the CPU wishes to have the services of the network interface card (NIC), it could signal the NIC using a pre-assigned IRQ line. By steering the IRQ to the TCBE's PCI INT number, the TCBE would receive the signal and its routines could then control the use of the NIC. [Ref. 14, p. 130] This method has not been tested in our lab.

c. Security Analysis of Personal Computer Interface Choices.

The PCI bus is the only viable solution for the PC to TCBE interface. The security implications of this must be explored. The capabilities of the PCI bus offer many potential problems. One problem that could cause unauthorized information flow is the fact

that devices on the PCI bus can transfer information without the benefit of the CPU. Another method of subverting a device on the PCI bus would be to re-map the PCI internal interrupt that belongs to the TCBE to another device. This could allow calls that are serviced by the TCBE be serviced by a physically adjacent device instead. These security problems can be alleviated by physically controlling access to the client PC. Additional devices are prevented from being added to the system that can communicate with the outside world. Additionally, the interface to the outside world will be controlled by the TCBE with high assurance. Thus, attempts to subvert the communications path would result in no unauthorized information flow.

5. TCBE Interface with the High Assurance Server (XTS-300)

a. Why an Interface with the High Assurance Server?

The primary duty of any interface between the TCBE and the High Assurance Server (HAS) is to establish the trusted path (TP). Subsequently, it would provide a path for the secure transfer of information from the HAS to the host PC. Access to information will be allowed in accordance with HAS security policies. The interface must support the requirements of establishing a trusted path. The TCSEC stipulates that trusted path communications and normal communications "...be logically and unmistakably distinguishable..." [Ref. 3] As a result, the method of interconnecting units in the LAN must support protocols for exchanging cryptographic keys and encrypted data transfer.

Protected channels are needed for both trusted path communications and protected session information transfer. The trusted path channel differs from that of the session, as it supports a high integrity protocol to provide proper identification of both sides of the channel. While data integrity is of some concern, the session channel is primarily used to keep communications secret from other clients on the LAN. It may also provide integrity so that information may be checked against modification and authenticity.

Control of the addressing is required by the TCBE because the interface with the protocol server on the HAS uses a non-standard port. Any e-mail service uses port 24,

Simple Mail Transfer Protocol (SMTP) uses port 25, and Internet Message Access Protocol (IMAP) uses port 143. Commercial software providers must use these well-defined ports. The system designed by S. Heller and S. Bryer-Joyner in [Ref. 12] uses non-standard ports to better control communications at the HAS. This requires the TCBE to change the port number from that given at the PC to that assigned by the HAS when the connection is made.

b. How to Interface with the High Assurance Server.

There are two possible methods for implementing the interface with the TCBE. One allows the use of a standard network interface card (NIC) that communicates using the Ethernet standard under the control of the TCBE through PCI interrupt re-mapping. This is an untested, theoretical method of control. Assume that there are two devices in adjacent PCI slots. The first slot contains the TCBE and the second has a NIC. Each slot is assigned a PCI INT number so the TCBE is INT#1, and the NIC is INT#2. The NIC is assigned an IRQ by the PC or the operating system. The PCI controller has registers that can be used to map an IRQ to a specific PCI INT number. In this case the PCI controller registers will be directed to send the IRQ signal meant for the NIC (INT#2) to the TCBE (INT#1) for servicing. In this way the TCBE can control the NIC's function within the client PC. As the TCBE is then placed between the operating system and the NIC, it can properly enforce the communications security needed in and out of the host PC. Proper application of cryptography, enforcement of trusted path communications, and routine session communications can be controlled. Figure 3 below shows the manipulation of PCI INT numbers their relationship with IRQs within the PCI Controller.

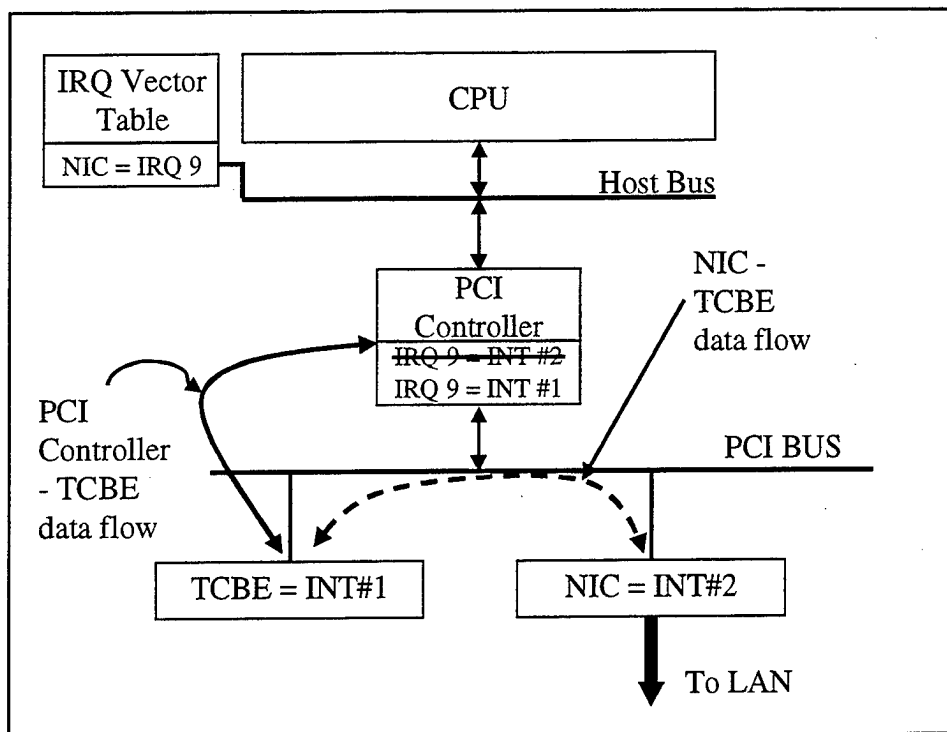


Figure 3. PCI Interrupt to IRQ Re-mapping

Figure 3 shows a potential method for controlling a PCI device (i.e., NIC) by “steering” its interrupt control handle (INT to IRQ reference) to another PCI device (i.e., TCBE).

The other method of implementing an Ethernet connection is to incorporate the NIC on the TCBE. Although this is a more difficult method of implementing the capability, it carries with it an assurance that the TCBE is mediating all communications to and from the host PC. The technology for implementing an NIC is well known and occupies a minimal amount of space as it may be accomplished in a very large-scale implementation (VLSI) device. As this could be the third capability added to the TCBE, power consumption must be monitored carefully.

In either case, a TCP/IP protocol will have to be implemented in the TCBE for trusted path communications as they are separated from the operating system that provides the TCP/IP protocol for all other communications. This is a rather important consideration in a

communications interface as the protocol stack must be carefully constructed and tested for weaknesses that have been discovered in other implementations.

c. Security Analysis of the High Assurance Server Interface.

There have been numerous attacks made on specific implementations of TCP/IP protocol stacks. These attacks have taken advantage of buffer overflows and denial of service that could occur with malformed packets. Although these attacks have been used against commercial operating systems and applications for them, there is little threat that improper information flow will occur as the TCBE is designed to prevent unencrypted information from leaving the host PC. Denial of service could occur, and be relieved by re-booting the system.

If the TCBE is attempting to control a NIC that is located in a separate PCI slot, a very real threat of uncontrolled unencrypted information transfer exists. If the NIC can be addressed by code running on the CPU without the TCBE's intervention, information transferred by the operating system/application can be exfiltrated. The implementation that exercises control over an external device must be thoroughly tested for weaknesses.

The NIC interface implemented on the TCBE exposes fewer vulnerabilities to attack from the host side. The host can see the TCBE as a NIC. This simplifies the abstraction between the host and the TCBE to that of a NIC. The driver used by the host operating system and applications is untrusted; the TCBE controls access to the LAN, thus mediating all communications to/from the PC.

6. TCBE Hard Drive Interface

a. Why a Hard Drive Interface?

On the TCBE, the purpose of the hard drive interface is to ensure that object reuse is properly carried out on the permanent media that must be used on the PC. It is impossible to remove the requirement for a hard drive from the Windows NT operating system without access to source code. It is therefore necessary to have the hard drive under the strict

control of the TCBE to ensure that the operating system is securely delivered to the PC and that the hard drive is properly cleansed between sessions.

The operating system must be delivered from a known source. This provides maximum assurance that there is no malicious code inserted into the operating system during one session and carried to another session. This does not provide greater assurance that the operating system, as delivered from the vendor, is doing exactly what it has been designed to do, but does provide assurance that each new instance of a PC session begins in a known configuration.

The reason for cleansing the hard drive between sessions is to control object reuse. The storage object is the hard drive and the potential reuse would be by a subsequent session. Cleansing, in this case, is overwriting the drive with an encrypted version of a fixed image of the operating system and applications that are available. This should sufficiently obscure any information that was stored in a previous session by the operating system and/or applications.

Finally, the greatest need to have a hard drive interface with the TCBE is that the method used to carryout object reuse must be completed with assurance. Either of the methods that utilize ghosting technology from Chapter III require that cryptography be applied. The methods that use ghosting technology must also be able to enforce control of read and write over an entire partition or drive (depending upon the method implemented, e.g., when a drive has not been readied in terms of object reuse, only the TCBE can read or write it.) This also to ensures that the image that is used to make the working partitions/drives remains unchanged by unauthorized sources.

b. How to Control a Hard Drive Interface.

Modern motherboards offer two channel Intelligent Drive Electronics (IDE) interfaces onboard. The IDE controller is built into the PCI/ISA bridge unit of the main controlling chipset. These controllers are accessible to the PCI bus as they are controlled by the PCI/ISA bridge, and they are fully programmable. This type of access is tenuous at best, and may not be sufficient because operating systems often access the hard drive directly vice

through the BIOS interrupt system. Bypassing the BIOS is used because it is more efficient. This is especially true with the modern FAST ATA and Ultra DMA models used in PCs today. The operating system's direct access will present a problem using this method of control as the TCBE must decrypt the information on the drive unit before the operating system can utilize it. Since the TCBE could only control access to drives driven from the motherboard by control of the BIOS routines, systems that don't use the BIOS routines for access can bypass the TCBE altogether.

Thus, there is only one sufficient way to obtain the degree of control necessary over the hard drive. The hard drive must be controlled directly through the TCBE by implementing a hard drive controller on the TCBE. As there are two types of drives common on the market, and each require different types of controllers, one might ask: "Which type of controller should we implement?" Intelligent Drive Electronics (IDE), and Small Computer Systems Interface (SCSI) are the two types of hard drives and controllers used most in industry. A two-channel IDE controller is native to the PCI chipset on most modern motherboards. This controller exists in one chip in the chipset. It is reasonable to speculate that the same type of controller can be implemented in a single chip on the TCBE and can therefore be achieved in a small footprint. SCSI disk controllers are general-purpose controllers and take a greater number of chips to implement. If the need in the system is for a high speed high capacity hard drive, then the SCSI implementation must be considered, however modern IDE hard drives rival the rapid data access capable on SCSI drives. IDE drives and their controllers are sufficient as the NPS MLS LAN is a server-oriented system, and there is little need for extremely fast high capacity hard drives.

An IDE controller can be implemented in a single chip, but the connectors are rather large and will take up some space on the TCBE card. Typical controllers can control only two channels each capable of controlling two IDE devices. This allows the control of four IDE devices in total. One of the devices will probably be a CD ROM. The CD ROM can be used to install and/or update the master image of the operating system and applications that are used in operating system delivery methods depicted in Figure 9 and Figure 10. This leaves

room for three drives and the ability to implement the operating system delivery method based on ghosting technology shown in Figure 10. The TCBE must be able to control one drive for read access only and use it as an image for the other two drives. The TCBE must then be able to generate keys and, using ghosting software made for the TCBE, copy the read-only drive's image onto the other two drives. This requires an interface with the hard drive, the cryptographic modules, and any memory that is maintained by the TCBE.

c. Security Analysis of the Hard Drive Interface.

Attempting to control access to IDE devices under the motherboard controllers is not sufficient. As noted above, operating systems often directly access controllers for greater efficiency. Since the TCBE's only mode of controlling the IDE devices is to re-map the BIOS routines that handle disk access, any access outside BIOS control will go unchecked. This violates the reference monitor design of the TCBE control over the hard drive, causes system instability as the operating system will attempt to read a device that is encrypted, and could cause corruption of the reference image on the read-only portion of the operating system delivery, as the read only status could not be enforced.

A hard drive interface on the TCBE is the only viable means of insuring that there is adequate control over the hard drive itself. This control is necessary for implementing the operating system delivery method, applying cryptography, and executing the ghosting program for overwriting the partitions/drives for object reuse purposes. The amount of code necessary to implement a hard drive controller is minimal and inspection for assurance purposes of the module is possible. The potential for subverting the hard drive from outside is minimal, as the code used to make up the TCBE will be protected from the operating system and applications. This method will likely require extra code in the TCBE BIOS software, as it must master the PCI bus to provide adequate data transfer rates. This extra BIOS code must be formed carefully and analyzed for vulnerabilities to attack by malicious code accessing it from the host.

C. CONCLUSIONS AND OBERVATIONS FROM THE EXTERNAL INTERFACE ANALYSIS

The implementations that attempt control of interfaces and functions outside the TCBE, offer a quicker development of a TCBE and the NPS MLS LAN prototype. The devices in adjacent PCI slots and on the motherboard can be controlled through BIOS manipulation and IRQ vector re-mapping to routines controlled by the TCBE. However, as seen in the case of the hard drive interface, direct access to controllers is often used by operating systems. It is equally probable that operating systems will attempt to directly access other devices, such as network interface cards, for the purpose of greater efficiency. This access is accomplished through the controllers on the motherboard that provide services to the CPU and cannot be mediated by the TCBE.

Moving the various controllers to the TCBE allows the TCBE to achieve the necessary control over operations to ensure the proper security of the client PC. The TCBE must be able to mediate all communications outside the host PC. The only way to accomplish this is to implement the NIC on the TCBE. It must be able to properly condition the hard drive for use between sessions. To do this, it must implement the hard drive controller on the TCBE. It must convincingly be able to communicate with the user for trusted path operations. To accomplish this, it must provide separately controlled input and output devices.

The ramifications of this are great. First, this means a more complex design for the TCBE. This will result in increased development costs. Second, the additional controllers take up space on the TCBE. Economy of space is required and is expensive. Third, there is a limited amount of power available to the to a PCI unit. The multiple device controllers and their supporting elements on a single device may consume more power than available. This concern again requires more expensive miniaturization and low-power technologies to reduce power consumption. Finally, this analysis mentioned, but didn't discuss the multiple support modules and internal interfaces that must be developed to securely implement the design. Included in these are: (1) cryptographic modules to perform calculations and generate keys, (2) memory modules to control memory allocation, protect storage of sensitive information

that must persist beyond a single session, and ensure the proper execution domain is maintained for software running on the TCBE, (3) BIOS control modules that properly handle the interrupt driven operations that are taken up on the TCBE such as, keyboard, video, network, and other internal processing (i.e., cryptography), (4) an internal bus structure to handle the many data manipulations required, and (5) a CPU with sufficient computing power execute the code needed to implement the TCBE. A functional block diagram may look the one below.

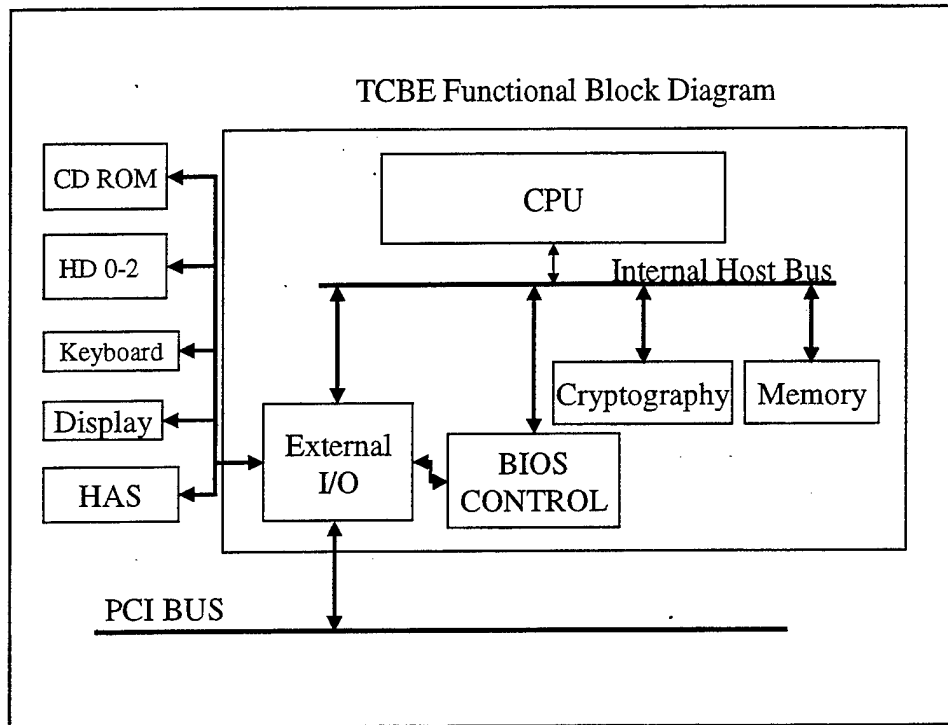


Figure 4. TCBE Functional Block Diagram

The block diagram in Figure 4 above attempts to summarize the interface analysis conducted in this chapter. The "External I/O" module has a considerable amount of functionality and responsibility. It is the sum of the interface control units described in this chapter. It is tied to the internal host bus to allow control by code in execution in the TCBE's CPU. It is also tied to the BIOS control module to allow access to BIOS routines and accept interrupt-driven communications. The Cryptography and Memory modules are shown for collateral duties of trusted path cryptography, session communication cryptography, IDE drive cryptography, and program execution for control of all these functions. The External I/O

module will be the "choke-point" in the design of the high degree of control and the large amount of data being handled by that module. It must be carefully designed and implemented.

THIS PAGE INTENTIONALLY LEFT BLANK.

III. DISK BASED OPERATING SYSTEM DELIVERY METHODS

A. A STATEMENT OF THE PROBLEM

The MLS LAN client operating system is Windows NT 4.0 Workstation. This operating system has certain features that require it to write to the permanent media from which it is loaded. This aspect of the operating system makes its storage and delivery into a high assurance LAN environment difficult. For instance, object reuse is a requirement for all systems of Class C2 and above. [Ref. 3] This requirement stipulates that the memory (RAM, cache, buffers, virtual memory, etc.) does not contain data left over from previous usage, which, in the case of a multilevel system, may be at a higher sensitivity level. For example, if a user opens mail from a high assurance server (HAS), the program used to open the mail might cache a copy of the mail in local memory. The cache may be flushed to secondary memory somewhere in the process. If the system were then shutdown, there would be information available on the hard drive. Subsequently, if a user who could only access the system at a lower classification were to log on and gain access to that drive space, information would be leaked. This information could be exploited. For this purpose, we must be careful how this operating system is delivered, and we must take measures necessary to ensure there is no data remanence from session to session.

A number of delivery methods for the operating system have been investigated. This chapter will enumerate these methods and discuss the advantages and disadvantages of each. Although the target operating system is Windows NT, these delivery methods are not operating system specific.

B. POSSIBLE OPERATING SYSTEM DELIVERY METHODS

Information stored on permanent media may be separated either physically or logically. Physical separation uses separate disks. Logical separation uses logical drives or partitions and/or cryptography. Windows NT, and many other operating systems, establish account and

hardware information upon system load, and utilize secondary storage for caching. This means that the operating system requires a permanent media that it can write to on each system load just to configure itself properly. For this reason, loading an operating system with the assurance that it has not been modified by malicious software inserted in a previous session is difficult. Furthermore, measures must be taken to ensure that information the operating system has stored in cache or in system configuration files is not leaked from high to low. Providing the client with a mechanism that takes care of these two problems is paramount to the success of maintaining security in the NPS MLS LAN.

Each operating system delivery method uses the common topography depicted in Figure 1. The type and content of information transferred by the high assurance server (HAS) will be modified slightly to support the use of cryptography and/or to deliver information to the TCBE needed to determine the initial conditions of the client device. The HAS will be the source of the identification and authentication (I&A) services, and depending upon the operating system delivery method and protocol, additional information needed to determine how to boot the operating system.

The general operational requirements for each analyzed operating system delivery method are: (1) the operating system should be delivered with as high an integrity as possible, and (2) the data that could be stored by the operating system should not be available during subsequent sessions at different classification levels. The first requirement is presented in an attempt to limit/eliminate the effect of malicious code that could be inserted into the operating system. The second requirement is presented as a means of ensuring that information is not leaked from "HIGH"-to-"LOW".

Each operating system delivery mechanism will describe:

- How the mechanism works,
- Its ability to satisfy operational requirements,
- An analysis of its security, and
- An analysis of its complexity, cost, and scalability.

One of the principle security aspects that must be identified is the delivery method's capability with respect to object reuse. Object reuse is well defined in the TCSEC.

Object Reuse: The reassignment and reuse of a storage medium (e.g., page frame, disk sector, magnetic tape) that once contained one or more objects. To be securely reused and assigned to a new subject, storage media must contain no residual data (magnetic remanence) from the object(s) previously contained in the media. [Ref. 2]

So, object reuse is the secure management of storage objects.

The purpose of object reuse is to prevent scavenging of information:

2.2.1.2 To prevent scavenging of information from previously used storage objects... The object reuse subsystem must perform its function for all reusable storage objects on the protected system (i.e., main memory, disk storage, tape storage, I/O buffers, etc.). [Ref. 17, p. 14]

The above quote further requires that the mechanism that enforces object reuse control all reusable storage objects under the control of the protected system. Below is the requirement that it be tamperproof and always invoked.

Object reuse subsystems must be interfaced with the protected system in such a way that they are tamperproof and always invoked. [Ref. 17, p. 14]

"All authorizations to the information contained within a storage object shall be revoked prior to initial assignment, allocation or reallocation to a subject from the TCB's pool of unused storage objects. No information, including encrypted representations of information, produced by a prior subject's actions is to be available to any subject that obtains access to an object that has been released back to the system." [Ref. 3, p. 15]

Of particular interest in the statement above is the fact that residual information that was in encrypted form, must be subjected to object reuse criteria before it may be reissued. Finally there are two acceptable methods for freeing up objects for reissue:

"...Information remaining in previously used storage objects can be destroyed by overwriting it with meaningless or unintelligible bit patterns. An alternative way of approaching the problem is to deny read access to previously used storage objects until the user who has just acquired them has overwritten them with his own data." [Ref. 17, p. 15]

Windows NT uses the latter method mentioned above. This can be verified through experimentation. First, shut down the NT system. Second, using a DOS boot disk with NTFS DOS – a program that gives access to NTFS disk volumes, and a common file viewer, one can load and observe the information within the swap file used by NT for caching. The information stored in this file depends on previous activity within the operating system, but is information not intended for un-restrained access. Forcing a core dump and then accessing the file that results allows access to information stored in system memory. This result justifies the need to bolster object reuse in the client through the TCBE actions.

There are some other common requirements for each operating system delivery mechanism. These are:

- Permanent Media Capacity

Each client PC hard drive or removable media must be capable of supporting one to two gigabytes of information per installation. This is to hold the operating system and any application software needed (e-mail browser, word processor, presentation software, etc.)

- New Technology File System (NTFS)

Each installation of the operating system and applications must be made on an New Technology File System (NTFS) volume. This is not an absolute requirement; it just facilitates securing the registry against the average user performing unauthorized changes to the operating system configuration. The Department of Defense (DoD) has established a Class “C2 configuration” of the Windows NT 4.0 workstation in a LAN. [Ref. 18] One of the first requirements is that the files system be NTFS. For this reason all test environments will be on NTFS systems.

1. Removable Media

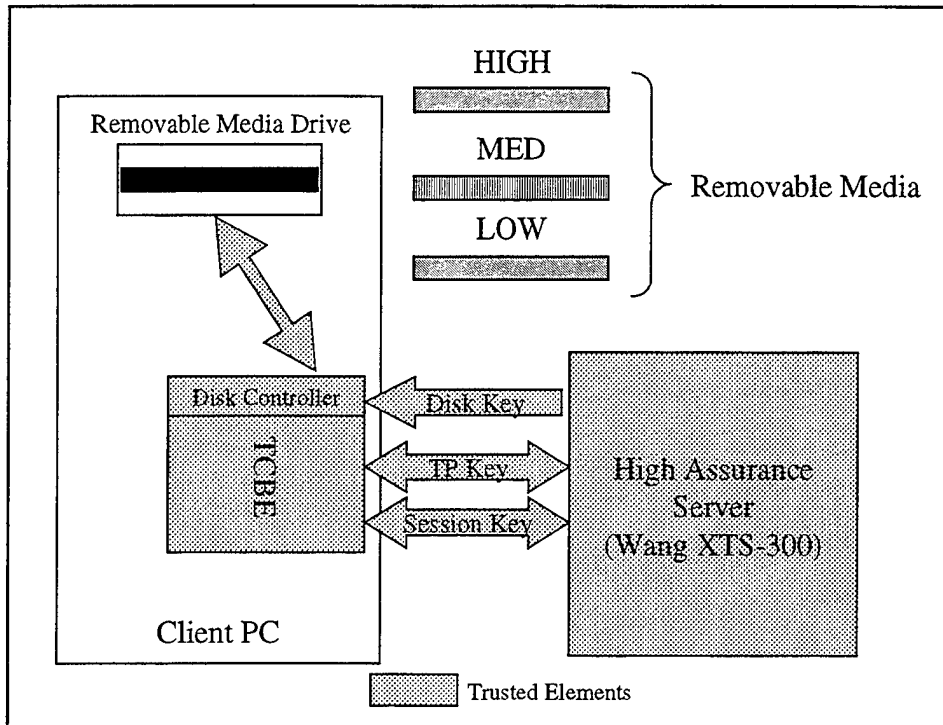


Figure 5. Operating System Delivery via Removable Media

The method in Figure 5 uses physical separation of data and cryptography to protect information. Each separately bootable disk is encrypted with a different key and assigned to a different classification level. The removable disk drive is under the direct control of the TCBE to improve assurance. The HAS manages all the keys.

a. Technique

An Iomega JAZ, or any other suitable removable hard drive, may be made a bootable device. The media may be encrypted to support the services at the selected session level alone and to ensure the user inserts the correct disk for the particular classification level. Any information stored on the disk is effectively classified at the level of the disk.

The user initiates a typical session by turning on the PC and/or pressing the Secure Attention Key (SAK). This activates the trusted path where the user carries out Identification and Authentication and session level negotiation procedures dictated by the HAS.

The HAS completes the trusted path communications by transmitting a session key and the appropriate cryptographic key for the session level negotiated. When this is completed the client PC is allowed to bootstrap the operating system. This is successful if and only if the appropriate disk is inserted into the PC, and the key for that disk was the one sent by the HAS. Without the correct key the disk is unintelligible. If unsuccessful, the user must insert the correct disk.

In order for this configuration to be practical, the keys used to encrypt the disks will need to be persistent. This requires some key lifecycle and a well-defined key management scheme. The HAS will potentially have to store one key per level per user. A protocol to describe how these will be accomplished is beyond the scope of this thesis.

The total number of keys required to secure this system, depends on the disk use policy. There are essentially two cases to consider. In the first case there is only one user per disk. In this case, the maximum number of keys are needed if there are m users and n classification levels, then there will need to be $(m \times n)$ keys and subsequently $(m \times n)$ disks. The number of actual keys (and disks) needed to operate the system could be reduced somewhat by the fact that not all users are able to access every level and these keys (and disks) need not be prepared.

In the second case policy allows multiple users per disk. This case relies on the discretionary access controls of the operating system to keep information separate between users on the same disk. The number of disks is reduced to the number of classifications that are accessible on the HAS plus any duplication of disks necessary to allow multiple users simultaneously at the same classification level.

This method does not secure the operating system against modification from session to session. There is little assurance that the original image that the operating system is made from is maintained after initial use. This does not present a confidentiality concern, but rather an integrity concern. If the operating system is modified and malicious code is inserted into the executive/kernel level of the operating system, information that is accessed by legitimate users could be modified without the user being aware. There are many other attacks

that could be perpetrated by such malicious code causing irreversible damage to all information within an access class. This damage would be limited to the access class by the HAS. For this reason it might be desirable to maintain the integrity of the operating system through some other method. For example, since the Windows NT operating system is made up of many files that load when the operating system is loaded, it may be possible to examine the checksums of certain static files before loading. This method will provide partial integrity only when the checksums are stored in a highly secure repository, viz. either on the HAS or the TCBE.

b. Advantages and Disadvantages

The use of removable media has as its major advantage simplicity. It is the easiest to install and configure. It requires an additional key exchange between the TCBE and HAS. As already stated, it fully supports confidentiality of data that is stored unintentionally by the operating system and applications. Another advantage is the fact that a user may personalize his/her copy of the operating system. Personalization could improve productivity as users could set up quick links and keys to particular applications. This method also supports an unlimited number of classifications, but each one supported requires a separate disk and this creates a financial and administrative burden.

There are several disadvantages to this approach. First, the number of keys that must be generated and maintained is large. It generates an administrative burden in preparing the disks for each of these keys. Second, this approach needs persistent keys. This requires that the choice of cryptography be appropriate for the type of material encrypted and the predetermined lifetime of the key. Of course any key that is compromised within its lifetime could cause a significant amount of damage. Third, the need to periodically change keys creates an unnecessarily high burden on the administrator. New removable disks must be made for each person/classification level in the LAN for each key that expires. All information encrypted with the old key must be re-encrypted with the new one. Perhaps automating the key change would reduce the burden on the administrator by shifting it to the user who must wait for the contents of his/her removable disk to be re-keyed. Fourth, the control required for each disk that is prepared creates a storage problem. Each user might require a safe that is

certified for storing these disks. Depending on local policy, there may be a requirement for separate safes for some of these disks. A fifth disadvantage is the overall cost of installation. The drive costs on average \$499.00. Each disk costs approximately \$129.99.¹ Sixth is the speed of the drives may affect system performance. They are typically much slower than the IDE drives used today. In a test conducted in our lab, a system was set up with a two gigabyte JAZ drive; the time from power-on to Windows NT desktop was three minutes. This is just start-up time. Configuring the system takes considerably more time. The files must be transferred from the CD ROM to the removable disk. This transfer is considerably longer than standard transfers between CD ROM and modern IDE hard-drives placed in typical systems today. During the configuration of the Windows NT operating system, there are several instances where a reboot is required. This again took considerably much more time than on a conventional hard-drive. This burden could be reduced by properly configuring a system, creating an image of the configuration, and then, using ghosting technologies mentioned in a later section and Appendix B, to copy the image onto the disk with the appropriate encryption applied. Finally, the system relies on integrity checks on static system files to detect tampering. This method fails if system files that are not checked are modified and able to cause damage.

Object reuse is improved by this implementation if each disk is limited to use by a single user. This is assured by the physical separation of the disks and the presence of only one disk in the PC at a time. Although this solution provides a solution for the object reuse problem, it introduces many other problems, which make it an unacceptable approach to object reuse.

¹ These prices derived from an Internet search on 7/24/99. They constitute an average of five on-line computer suppliers. They do not consider volume discounts or government rates.

2. Large Permanent Disk with Multiple Partitions

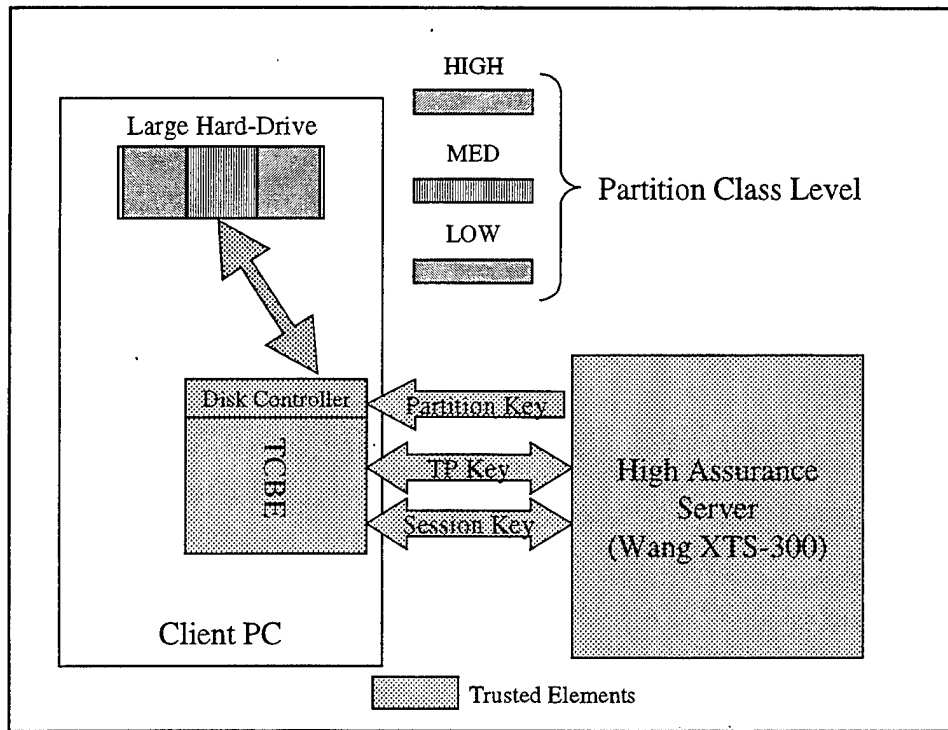


Figure 6. Large Disk with Separate Partitions for Each Level

The method depicted in Figure 6, uses logical separation for the protection of data using partitioning and cryptography. The keys are provided by the HAS for client/TCBE communications with the HAS and for encryption of the partitions on the hard drive. The hard drive controller is incorporated into the TCBE to improve assurance. Keys and partition information are maintained by the HAS.

a. Technique

One large disk with multiple partitions can be used to deliver the operating system. Each partition will have to be large enough to support a full installation of the operating system and the applications - about two gigabytes each. Each partition has an identical installation of the operating system and the applications chosen for use in the office with the exception that each partition is to be cryptographically locked to ensure that the data it contains are not accessible when sessions at other access classes are running from other

partitions. Each partition could be controlled per-user or per-access-class. Under per-user controls, each user has a partition at each access class for which the user is authorized. Per-access-class controls ensure that the partitions are configured per classification and users must share the operating system and applications for each classification level. The cryptographic keys for encrypting the partitions and the partition information for each classification level will be stored on the HAS. The master boot record (MBR) information needed to determine which partition to boot will be loaded from the HAS along with the appropriate encryption key. This information will be placed into a designated location in the TCBE for later use. This is required to prevent the TCBE from needing to determine policy.

A session begins in a similar fashion as the case above with the user turning the PC on and/or invoking the SAK. The Identification and Authentication with the HAS is completed and the session level is negotiated. The HAS delivers the session key which is needed for encrypting data that is transferred to and from the HAS along the LAN, a partition unlock key which is required to successfully boot and run the operating system and applications that have been prepared for the negotiated classification level, and the MBR modifications that will cause the appropriate partition to be used. Then the TCBE modifies the MBR so that when the operating system is allowed to load, the MBR points to the correct partition for the session level negotiated. Finally, the operating system is allowed to boot.

The number of partitions required depends on the policy for use of the client PCs in the LAN. If each PC is designated to an individual, then the number of partitions required is limited to the number of classification levels that the particular user has access to. If each PC is designated for general use, and partitions are shared, the number of partitions is also reduced to the number of classifications accessible to the system. If partitions are not shared, then the number of partitions increases considerably as there must be a partition per user per access classification the user can access. In particular the maximum number of partitions required for m users and n classification levels is $(m \times n)$.

Integrity can be improved beyond the capabilities of the operating system by creating checksums for static portions of the operating system. These checksums must be stored securely and compared before operating system load.

b. Advantages and Disadvantages

The principle advantage to this configuration is the ease of setup under Windows NT. Windows NT allows the set-up of multiple versions of itself, each existing on different partitions. The partition on which the operating system resides need not be a primary partition. Since the operating system allows up to four primary partitions or a combination of primary and extended partitions, a drive may be configured with essentially an unlimited number of logical drives. For instance a fourteen billion byte (gigabyte) hard drive may be partitioned to handle seven two gigabyte logical drives in the following way: One two-gigabyte primary partition, one twelve-gigabyte extended partition. The twelve-gigabyte extended partition can have three two-gigabyte logical drives and one six-gigabyte extended partition. This six-gigabyte extended partition can be split into three two-gigabyte logical drives. Then the total number of logical drives each two-gigabytes in size is seven. This would theoretically hold seven separately encrypted security levels or four traditional security classifications and a number of compartments for each. By partitioning the hard drive into partition sizes closer to the minimum sizes needed, the number of available partitions could be maximized. As stated earlier, it provides good protection of data that is unintentionally stored by the operating system and applications during a session at a particular level.

The principle disadvantage is the lack of scalability. There is a real upper limit to the number of classes and/or compartments that could be placed on one drive or to the number of users that a particular client PC can support. Another disadvantage is the need to keep keys for this configuration. Key lifetime presents a problem of a different sort. Each time a key expires and must be replaced, the partition encrypted with the expiring key must be recovered and re-encrypted with the new key. This creates an administrative difficulty. This solution presents a single point of failure. Any drive failure creates a need to replace all partitions. This administrative burden could be mitigated by consistent back-ups on a device

with a system high classification. Restoration would be a trusted process. If a new classification is required and there is no extra room on the disk, an old partition must be overwritten.

This method improves object reuse issues in systems that have separate partitions per user per classification. Proper object reuse is assured by the logical separation between partitions and the separate encryption assigned to each partition. The other disadvantages imposed by this method make it undesirable as a solution for object reuse in a MLS LAN.

3. Multiple Permanent Disks (One for Each Level)

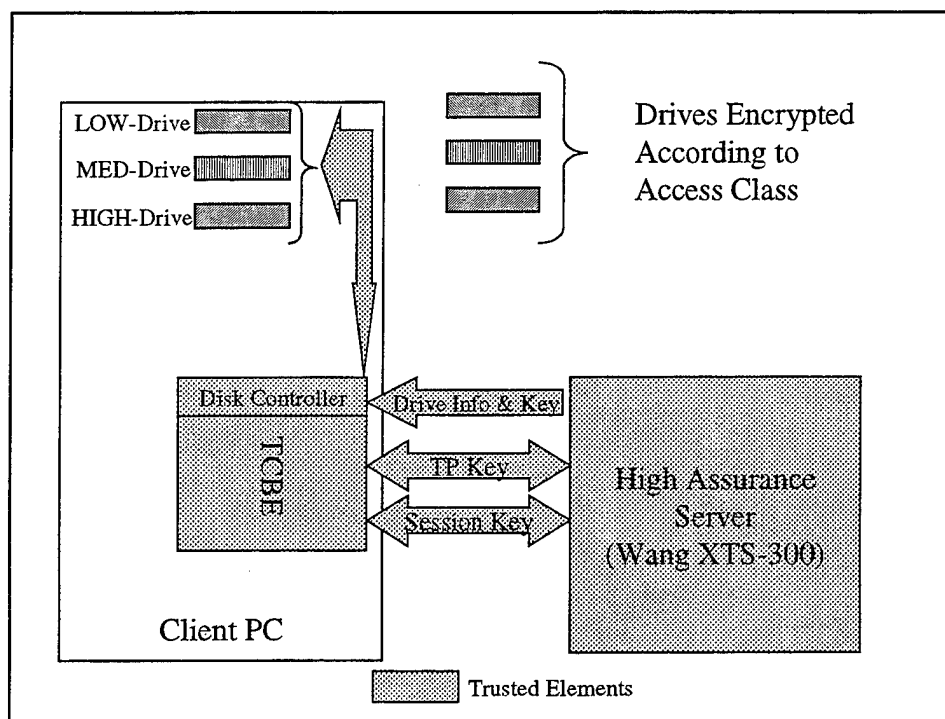


Figure 7. Separate Disks for Each Classification Level

The method in Figure 7 uses physical separation for the protection of data. It is the physical extension of the multiple partition method mentioned above, but is much more limiting due to space and power constraints. The hard drive controller that is part of the TCBE controls each disk. Key and drive information are maintained by the HAS.

a. Technique

This client will require a disk for each sensitivity level supported by the MLS LAN that the user is authorized to access. On each disk, an instance of the operating system and the required applications is loaded. Each will be cryptographically locked against the scavenging of data from one drive to another while a session is running. There are some differences between the case with multiple partitions and multiple disks. Where the active partition and MBR were modified for the partition case, the Basic Input/Output System (BIOS) setup and CMOS data will be modified to determine which drive to boot. The proper drive and the correct key will have to be sent from the HAS to the TCBE when the session is negotiated after logon. The system operation is similar to the previous two cases, except the information that is sent to the TCBE after successful session negotiation is: (1) Session key – for communications to/from the TCBE to the HAS, (2) Disk cryptographic key – to access the appropriate disk drive unit for the access class negotiated, and (3) the disk drive identification number to determine which drive to access.

The typical PC has two Intelligent Drive Electronics (IDE) channels one primary and one secondary, that can each support two IDE devices one master and one slave, for a total of four devices. Additional IDE controllers may be installed each capable of controlling four or more hard drives each. Typical Small Computer Systems Interface (SCSI) controllers are capable of controlling seven devices each. If the TCBE directly controls the hard drive by providing the controller onboard, the limited space of a single add in card that has many other functions will place a rather small upper limit to the number of drives that it can support. Typical power supplies have ten or less power couplings. This also places an upper limit to the number of devices that require power that can be installed into a single PC. Finally most desktop PCs or mini-tower cases have five or six empty drive bays. So it is easy to see that the number of classifications supported by each PC is severely limited. In most cases there will be two IDE channels available for a total of four devices. There will be a need for a CD ROM for software installation and controlled updates to existing software configurations. This

leaves room for three additional IDE devices. Most likely these will be three hard drives for storing operating system and application images.

As in the previous cases, this method does little to ensure the integrity of the operating system from session to session beyond the native capabilities of the operating system itself. Legitimate modifications are possible if users are allowed to personalize installations, and in any case, the operating system will modify itself on each boot. So, additional measures may be desired to improve integrity of the system so that access class information is not destroyed by malicious code that could be inserted into the operating system. Cryptographic checksums of important static files in the operating system may improve integrity.

b. Advantages and Disadvantages

This solution is presented solely as a stepping-stone for the next method of operating system delivery – Multiple Permanent Disks with Multiple Partitions.

Advantages of this method are few. It is an adequate means of preserving confidentiality by taking advantage of the physical separation of disks and the use of encryption. The amount of additional information that must be stored by the HAS is small – disk drive keys, and a disk-to-access class mapping. Note that the mapping constitutes a policy-relevant database so this solution requires the HAS to store the mapping and deliver the relevant information to the TCBE vice make the TCBE decide which drive to boot.

Disadvantages, on the other hand, are many. The number of keys that must be controlled raises an administrative burden. The use of persistent keys requires that the cryptography used must be stronger. This could raise the computational requirements of the TCBE. The management of keys with finite lifetimes is another administrative burden, as with each key expiration, a drive will need to be recovered and re-encrypted. The limited number of disks available on each PC creates a severe limit to the number of classifications supported on the PC. If policy is such that personnel with the same clearances may share disk drives, then the number of drives necessary is reduced. If policy requires that each individual have separate drives per access classification authorized, the number of drives and the subsequent administrative and financial burden is increased considerably. BIOS and CMOS data

modification are required prior to each bootstrap of the system. It is possible to accomplish, but it is difficult to complete in a timely manner, as it requires invoking the BIOS's CMOS setup routines to modify the required information and then save data and reboot. This also requires that the BIOS be left in a state that allows it to be modified. This opens the system to a form of virus attack that could render the machine useless. The attack, similar to the one enacted by CIH/Chernobyl rewrites a crucial part of the BIOS signature that makes the PC unbootable. This condition is unrecoverable if the BIOS is a soldered Flash ROM connected to the motherboard, as it is destructive to de-solder the Flash ROM chip from the motherboard to replace it.

Object reuse is improved by this method if system policy allows only one user per disk. Object reuse is assured by the physical separation of the disks and the use of cryptography that ensures intelligible access is limited to one disk at a time. The severe limitations presented by this solution make it undesirable as a solution for object reuse in the MLS LAN.

4. Multiple Permanent Disks with Multiple Partitions

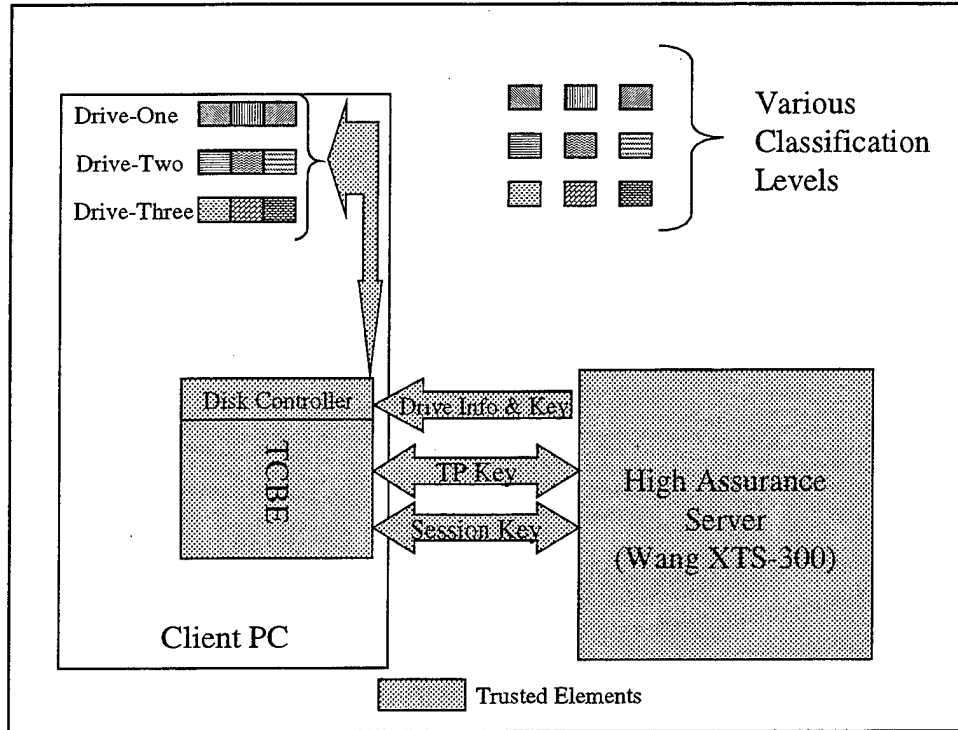


Figure 8. Multiple Disks with Multiple Partitions

The method in Figure 8 is an extension of the previous means of delivering the operating system to the client. It utilizes both logical and physical separation of data, and encryption to enhance confidentiality. The hard disks are controlled by the TCBE to provide better assurance that separation is maintained. Although limited, it presents a marked improvement in the number of classification levels accessible to a single client PC in a MLS LAN. Drive, partition, and all key information is maintained by the HAS.

a. Technique

This method presents a means of using multiple disk drive units each configured with separate partitions. Separate cryptographic keys secure each partition. The method of operation combines the single disk with multiple partitions and the multiple disks with single partition scenarios. The user would turn on the PC and/or invoke the SAK to begin Identification and Authentication procedures to be followed by session level negotiation. Upon

successful completion of the session level negotiation, the HAS will transmit a session key to be used for the secure transmission of data to and from the TCBE and the HAS. The HAS will also transmit the necessary partition key and partition number. These will be used to unlock the operating system and applications stored on a particular partition. Finally the HAS will send the disk drive identification number for the drive unit that holds the selected partition for the access classification that has been granted. The disk drive unit will be used by the TCBE to modify the BIOS and CMOS information so that only that disk is bootable. The partition information will be used by the TCBE, as it must modify the MBR to make the particular partition the active partition. Making a partition active makes it the boot partition for a particular drive.

b. Advantages and Disadvantages

As this method is a combination of the previous two operating system delivery methods, it provides a combination of advantages and disadvantages. One advantage is that it provides an adequate means of preserving confidentiality by taking advantage of both the physical and logical separation of disks and of the use of encryption. The amount of additional information that must be stored by the HAS is small and includes partition keys, and disk/partition-to-access-classification mapping. This method presents a marked improvement in the ability to support numerous access classifications. It is the easy to setup under Windows NT as Windows NT allows the set-up of multiple versions of itself, each existing on different partitions or drives.

Although there is an improvement in the number of access classes that can be maintained on this system, the total number of access classes is still quite limited. It is not as scalable as might be required in an organization that handles multiple access classifications. This method maintains the disadvantages of requiring keys for each user and user level. The presence of persistent keys requires the careful choice of cryptography appropriate for the type of information being encrypted and the predetermined lifetime of the keys. The management of key lifetimes is still a problem. All of these factors create an overwhelming administrative difficulty. Any drive failure creates a need to replace all partitions and operating systems on

that drive. With multiple drives, the administrative burden is amplified. This burden could again be mitigated by consistent back-ups on a system-high device.

The manipulation of BIOS and CMOS still presents technological, security, and social problems. Technological problems come from the dependence on the chipset of the motherboard in question for the ability to modify the BIOS and CMOS information. Security problems exist because the BIOS must be left in a state that allows it to be modified. Current methods to change the drive boot order stored in CMOS data require the use of BIOS setup routines. A machine reset is needed to access these routines. Another machine reset is needed after saving changes to the CMOS data so that the information that has been modified may be accessed. Two machine resets present a user acceptance problem because of the time it takes between sessions. Developing a way to change CMOS data without using the BIOS setup routines, thus eliminating the extra machine resets, may mitigate this problem.

This method does not protect the integrity of the operating system or applications beyond the inherent capabilities of the operating system itself. An improvement could be obtained by cryptographically hashing key files to detect any manipulation by hostile entities.

This method improves object reuse in the case where there is one user per partition. In this case physical and/or logical separation coupled with cryptography ensure that information cannot be accessed from adjacent partitions.

5. Disk Mirroring/Ghosting Applied to a Single Permanent Disk

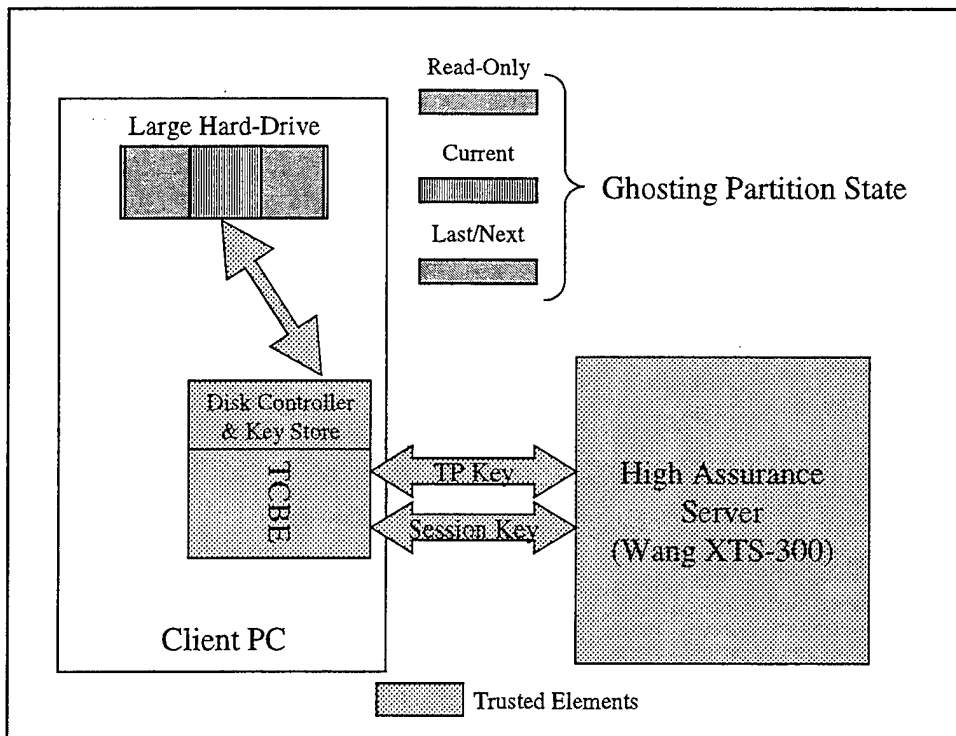


Figure 9. Ghosting Applied to a Single Disk with Multiple Partitions

The method in Figure 9 utilizes logical partitions and encryption to separate per session information. The TCBE controls the hard drives and runs software to copy new images of the operating system/applications over previous images to provide for object reuse on the permanent media. In this case the TCBE may store keys for the partitions, but the HAS stores the trusted path and session keys. One partition is read-only. The other two are encrypted to maximum capabilities.

a. Technique

This method of operating system delivery requires one six-gigabyte (minimum) hard drive partitioned into three partitions of at least two gigabytes each. The first partition is read-only and is used as a master image of the operating system/applications to be placed on the other two partitions in encrypted form. The other two partitions are used alternately as the "current" partition and the "last/next" partition. The TCBE must know the status of each

partition maintaining a "dirty" bit and an "in-use" bit for each. If a partition's dirty bit is set, it is either in use or has been used, and must be recycled when it is no longer in use. If the dirty bit is not set, it is ready for use. Setting and clearing the status bits must be trusted as reuse of a partition could cause an unauthorized information flow. The fail-safe should be for the dirty bits to be set regardless of the in use bit status requiring new partitions to be prepared.

The system is started as in previous cases. The user turns on the client PC and/or presses the SAK. The HAS directs Identification and Authentication procedures with the user. When this is complete the session level is negotiated. If a new session level is negotiated, the HAS informs the TCBE. After successful session negotiation, the HAS provides the session key for communications between the client/TCBE and the HAS. At this point the TCBE is free to start the operating system if one is encrypted, on the hard drive, and ready. From this point two cases are possible.

The first case assumes that there are no partitions in a state with both in-use and dirty bits clear. The TCBE senses that no partitions are ready and generates a key for encrypting the new partition. This key will be capable of securing the highest access class in the trusted server. Then the TCBE copies/ghosts the first partition onto the second, encrypting it with the new key, thus creating an exact replica of the first partition in encrypted form. At this point the client PC has one partition ready and the TCBE can start the operating system to begin normal session operations. During normal operations, the TCBE sets the in-use and dirty bits of the current partition, and directs the preparation of the other partition. The TCBE then generates another unique, key for the unused dirty partition and then copies the image of the first partition onto the third partition as a background activity. When this is complete the TCBE clears the dirty bit for this partition so that it can be used next.

In the second case, one partition is ready and the other is dirty and neither is in use. The TCBE tests the in-use and dirty bits, and finding a partition with neither bits set, starts the operating system on that partition. The TCBE sets the dirty and in-use bits for that partition and finds the partition that is dirty but not in use. It prepares a new key for this partition and copies the static partition onto it in encrypted form. Finally the TCBE clears the

dirty bit for the next partition so that it may be used later. This case represents the expected normal condition of the client PC. While a session is in progress, to mitigate the time expense of preparing a partition for use, the TCBE prepares another partition for future use as a background activity.

b. Advantages and Disadvantages

This method presents numerous advantages. First, information confidentiality is maintained because the cryptographic algorithms and keys used are suitable for the most sensitive information that could ever be stored on an active partition. The base partition is read-only and "in the clear", but the operating system cannot be started from a read-only image as it must be able to write to the partition that it is started from. Second, the amount of information about the client PC that must be stored on the HAS is reduced. There is only one disk to boot, and there are no keys to transfer for unlocking partitions or disks as they are generated and stored by the TCBE. Third, this method is the first to support object reuse by overwriting before reissue. Each partition will be overwritten before it may be used. This successfully fulfills the requirements given by the TCSEC for object reuse by overwriting with unintelligible bits. Fourth, this method is the first to support improved operating system integrity. The read-only partition stores the operating/system in a known initial condition. It cannot be modified as it is read-only and under the control of the TCBE. Since the operating partitions are created from this image, before every use, they also begin from a known state. This improves initial integrity on the client PCs. Any attack by malicious code on operating system integrity at the client could last at most the duration of one session. (Of course, that individual could corrupt the entire access class by sending e-mail with malicious executables in the attachments thus propagating the damage.)

There are a few disadvantages to this method. The first and most striking is the requirement for the TCBE to hold state the of the partitions. Second, the TCBE stores keys that must be saved even with the power removed from the TCBE by the PC. Both of these mean that there must be a persistent store, Flash ROM, or battery backed RAM. This complicates the design of the TCBE considerably. It might be possible to move both the

responsibility for storing keys and state of the partitions to the HAS at the cost of a more complex protocol for passing the information and modifying the HAS to store that information. Third, this method ghosts the read-only partition while the operating system and applications are running on the main CPU. This means that the disk will be accessed frequently and/or the TCBE will be busy communicating with the outside world. This means that the ghosting process, since it uses a single channel of the IDE controller will carry with it an overhead that may slow down the rest of the system. This can be alleviated somewhat by placing the active partitions on separate drives. This should reduce the overhead and communications bottleneck at a single IDE controller channel. Fourth, this method does not support the personalization of the operating system. The images are made from a standardized copy, and any changes that are made on the working copies are done away with when the partitions are re-written. This might cause a reduction in productivity, but may be considered acceptable in light of the increased security. Finally, the software that was used for testing is the commercial program Ghost made by Symantec. This program runs only in DOS. A program with similar capabilities must be developed use on the TCBE. The TCBE must be fully in control of the entire process to provide the high degree of assurance required.

6. Disk Mirroring/Ghosting Applied to Multiple Disks

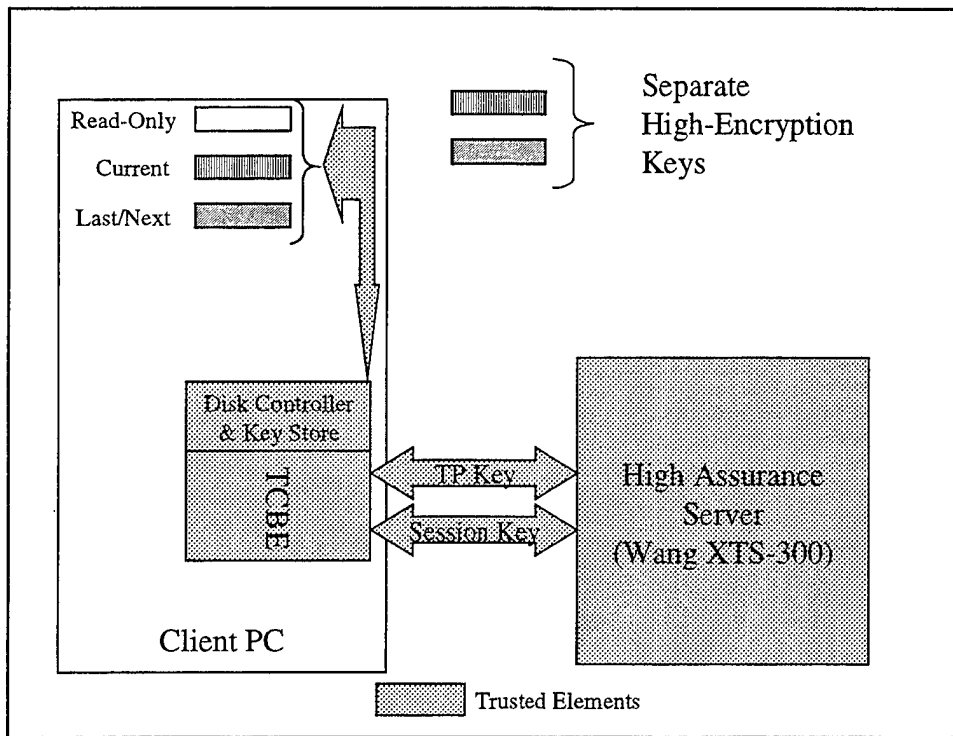


Figure 10. Multiple Disks with Ghosting Technology

The method in Figure 10 protects data based on the principle of physical separation and encryption for securing information. The TCBE controls the hard drives and runs software to copy new images of the operating system/applications over previous images to improve object reuse on the permanent media. In this case the TCBE may store keys for the drives, but the HAS stores trusted path and session keys. One drive is read-only. The other two are encrypted to maximum capabilities.

a. Technique

This method is very similar to the previous case. It differs in that there are three drives used instead of three partitions. The first drive is a read-only image of the desired operating system and the applications configured to run on that system. The other two drives are used for normal operations alternately for each session established with the HAS. The TCBE will need to keep track of the state of the two operational drives. The TCBE would

control bits that are set if the drive is "in-use" and "dirty". A dirty drive is one that has been used but not overwritten. If a drive is dirty and not in use, the TCBE must create an encryption key and make an encrypted copy of the read-only drive onto the dirty partition. If both drives are dirty, the TCBE must prepare at least one drive before an operating system can be started. To ensure the confidentiality of the information on these drives/partitions they would be encrypted using cryptography capable of protecting the most sensitive information in the system. This is needed to prevent the transfer of information from the current drive to the next drive. The operating system will see the next drive as being corrupt due to it being encrypted with a separate key. To prevent the current session drive from tampering with the next session drive, the TCBE should hide the presence of the next drive from the system. If this cannot be accomplished, a check sum should be calculated on the next drive and stored by the TCBE in persistent memory. Before the next drive can be used, another check sum would be calculated. If the two check sums do not match, the drive has been tampered with and should be recopied. Drive keys could be transmitted from the HAS or the keys could be generated by the TCBE, thus simplifying the protocol. New keys would be generated each time the source image is copied/ghosted onto the working drive further improving security of the system.

As seen in the analysis of the multiple disk method, three disks could be available; we can divide the responsibilities and maximize performance. The Primary Master drive could be configured as a read-only (read/write under the complete control of the TCBE for the purposes of operating system administration and update). This drive will store the desired image of the operating system and the applications. The other two disks will alternate duties as the current disk and the next/past disk. The current disk was prepared in the background during the last session. It was encrypted using a key capable of protecting the most sensitive information on the LAN. While the operating system is running the current session, the TCBE will prepare the next/past disk for future operation. A new key will be calculated and assigned by the TCBE. The disk image will be made from the Primary Master drive that remains unchanged.

The Primary Master drive must be bootable. This means that it must have a valid Master Boot Record (MBR), and the files needed for the operating system to initially load. For Windows NT 4.0 these files are NTLDR, NTDETECT.COM, and BOOT.INI. NTLDR is the code that loaded and executed by the MBR boot loader. It controls the operating system selection process and hardware detection prior to kernel initialization. BOOT.INI contains information about the location of the operating system to be loaded. NTLDR displays this information and accepts the user input for which operating system to load or just loads a default operating system declared in BOOT.INI. The TCBE will be required to modify BOOT.INI so that the correct operating system version is loaded from the current disk. To boot the system, the drive that is ready and not in use is the candidate session drive. The TCBE determines this drive by examining its in-use and dirty bits. It modifies the BOOT.INI file on the primary drive to point to the candidate session drive and loads the appropriate cryptography for the candidate drive. When the MBR boot loader executes, it loads NTLDR and BOOT.INI, parses BOOT.INI for the correct drive to boot, and then loads the operating system from that drive. This is ensured by the fact that the correct cryptography makes the correct version of the operating system readable and thus loadable.

b. Advantages and Disadvantages

Advantages of this approach abound. All the advantages of the previous example are repeated and a new one is added. This method mitigates the time and overhead required for copying the read-only image onto the working drives by allowing the drives to be run on separate IDE channels. This allows one drive to run for the operating system and the other to be re-written. The keys need only be kept for two drives. They can be generated locally and be stored in tamper-responding memory. In experiments it took only three minutes to prepare a drive that was configured with Windows NT version 4.0 and Service Pack 5, Microsoft Office 97, and Microsoft Internet Explorer 5.0. The ghosting software allows a drive to be prepared over a network (although securing this operation on the MLS LAN is likely to make it a much less rapid operation.) The drives in the experiment were not the same size and the operation succeeded. The only requirement is that the target drive have sufficient

capacity for the operating system and application image. If three drives are being used and one is a read-only master drive, preparing one drive in background would ensure that a drive is available in three minutes. This means no appreciable delay for normal operation.

There are a few disadvantages to this solution. One is the possibility that a drive may not be ready before a user completes logon. The maximum wait time is three minutes assuming no other contributing factors. The other disadvantages of the previous case are still present. The choice of the key source, key storage facility, and the storage place for the state of the drives all have implications. Choosing the TCBE as the locus of this activity requires a more complex and expensive TCBE, while choosing the HAS requires extending databases on the HAS and creating a protocol for passing the information. Finally, fact that a drive must be made alternately read-only and read/write by the TCBE requires careful implementation and evaluation for covert channels.

C. GENERAL COMMENTS AND CONCLUSIONS

The last two methods are best because they provide the greatest secrecy, integrity, and support for object reuse. Object reuse is supported because the disks that are used for running the operating system and applications are cleansed before every session. Security is improved by the fact that the keys are generated locally and that they exist for only one session. Integrity is improved by the operating system and application images being stored on a read only media and then flashed to a working disk before booting. In this way, the malicious code cannot affect either the operating system or the applications prior to their initial load. If all other aspects of the TCBE are designed well, malicious code could only enter the working environment during a session and, for the operating system and application source, will be purged between sessions. The effect of malicious code would be limited to the same damage that could be accomplished by migrating malicious code into the HAS. This shows the effectiveness of the TCBE in supporting policy enforced by the HAS TCB.

IV. USING WINDOWS NT 4.0 TERMINAL SERVER EDITION

Windows NT Terminal Server Edition provides many attractive features that might be useful in the context of an MLS LAN. It offers improvements in configuration management. Updates to the operating system and applications served by the system are centralized and offer a single place for updates. The Windows NT GUI is deliverable to cheaper PCs in that they may be diskless, have less memory, and need not run on state of the art, high-performance CPUs. The diskless client PCs prevent the storage of information permanently on the user's desktop, a marked security improvement.

In this chapter, possible configurations of a Windows TSE-enhanced MLS LAN are considered. For each configuration, a sketch of the topology and possible protocols are provided. Analysis of each configuration yields its advantages and disadvantages. First the basic features of Windows NT Terminal Server Edition are presented.

A. OVERVIEW OF WINDOWS NT 4.0 TERMINAL SERVER EDITION (TSE)

TSE offers a multi-user environment with lower administrative cost than fully functional PCs connected in a traditional client/server environment by offering a single source for operating system and application configuration and maintenance. Server requirements for TSE are one hundred twenty-eight megabytes of free hard drive space and thirty-two megabytes of RAM plus four to twelve megabytes of RAM per user. It is estimated that a single processor server can support fifteen to forty-five clients dependent on the user's usage profile. User usage profiles are a way of classifying the type of processing typically done by a user and are split among three classifications, light or task-oriented user, medium or administrative user, and heavy or knowledge user. The task-oriented user typically has one application open at a time working on single documents. Administrative users are average users who may be creating small documents and using e-mail or browsing the web. Heavy users work with documents that are more complex and have heavy use of e-mail and web services.

TSE is capable of delivering a Windows NT 4.0 graphical user interface and work environment to PCs that are not otherwise capable of running Windows NT themselves. TSE uses a type of terminal emulation to accomplish this. Terminal emulation is the process of capturing a computer's (terminal's) keyboard strokes and mouse events and transmitting them to a separate computing device for processing. The processing device then returns changes in graphics displayed on the PC's (terminal's) screen. Some client PCs may not be able to run Windows NT because they do not meet minimum requirements for running the operating system. Current Minimum requirements for Windows NT are:

- A personal computer with Pentium or faster processor,
- 16 MB of memory (32 MB is recommended),
- A hard-drive with 110 MB of free space for typical installation,
- A CD-ROM or access to CD-ROM over a network,
- A VGA or higher resolution display adapter, and
- A Microsoft Mouse or compatible pointing device. [Ref. 19]

Another reason why PCs cannot run Windows NT might be due to the use of legacy Windows environments, or other operating systems such as UNIX or Macintosh within the network. Of course, one might wonder why an organization might use TSE when many of its PCs are capable of running Windows NT. TSE provides a unique benefit from the fact that updates and management of the operating system are centralized. This is a great advantage for an administrator. It also centralizes applications that would be common to every user on the network. TSE is also capable of delivering the operating system to the rising market of thin clients and network PCs. These devices may not have a hard drive, but typically load a specially modified Windows compatible operating system (such as Windows CE) from read only memory (ROM).

TSE comes with three components. A multi-user server core is loaded onto the PC that will act as the server. Another component is the remote desktop protocol (RDP). RDP

connects the server to the client over TCP/IP network. It is an extension of the ITU T.120 protocol used for multi-channel conferencing. [Ref. 20] RDP is capable of making over 64,000 virtual connections with end-to-end encryption of 40 or 128 bits per connection. There are two versions of the client portion: one sixteen bit version for PCs running Windows For Workgroups 3.11 and the other a thirty-two bit version for PCs running Windows 9x and Windows NT. Third party software is required for PCs that do not run Windows in any form (UNIX, Macintosh, DOS, etc.)

As stated above, there is client software. Terminal Server's client software is suitable for Windows-based computers. It must be loaded onto the computer, so each PC that will act as a terminal must have a hard drive (typically four megabytes of free space), and operating memory (four to sixteen megabytes depending on the operating system). PCs running non-Windows operating systems must have third party software installed. Citrix Metaframe uses the Independent computing Architecture (ICA) protocol and may be used to connect DOS, UNIX, OS/2 Warp, JAVA, and Macintosh PCs as well as some Network Computers to TSE. The Citrix client also supports all Windows based operating systems. It is able to communicate via TCP/IP, IPX, SPX, NetBIOS, and asynchronous (modem) connections to Remote Access Services (RAS) used by remote offices to contact home offices. PCs that already have an operating system loaded and either the TSE client or Citrix client, may be configured either to boot to the native operating system or to initiate a TSE session. If running the native operating system, they may initiate one or more TSE sessions simply by clicking an icon. Diskless thin clients and network PCs that are RDP-enabled may be directly attached to TSE. Thin clients and network PCs may support the ICA protocol. These are connected to TSE running Metaframe, a Citrix product that makes it easier to deploy, manage and access the enterprise by providing TSE with management tools and the ICA protocol. Though these PCs are diskless, they have moderate computing power, and boot their operating system from ROM. Neither of these diskless connections require client software, but have specially modified operating systems that allow the connection to TSE. The principle purpose of the client software and the RDP- or ICA-enhanced network PCs and thin clients is to capture the keystrokes, mouse

movements, and mouse clicks and send them to the TSE server, and to transmit Windows objects from TSE to the client display. Those clients with multimedia enhancements also support sound.

Below are two possible topologies for TSE based networks.

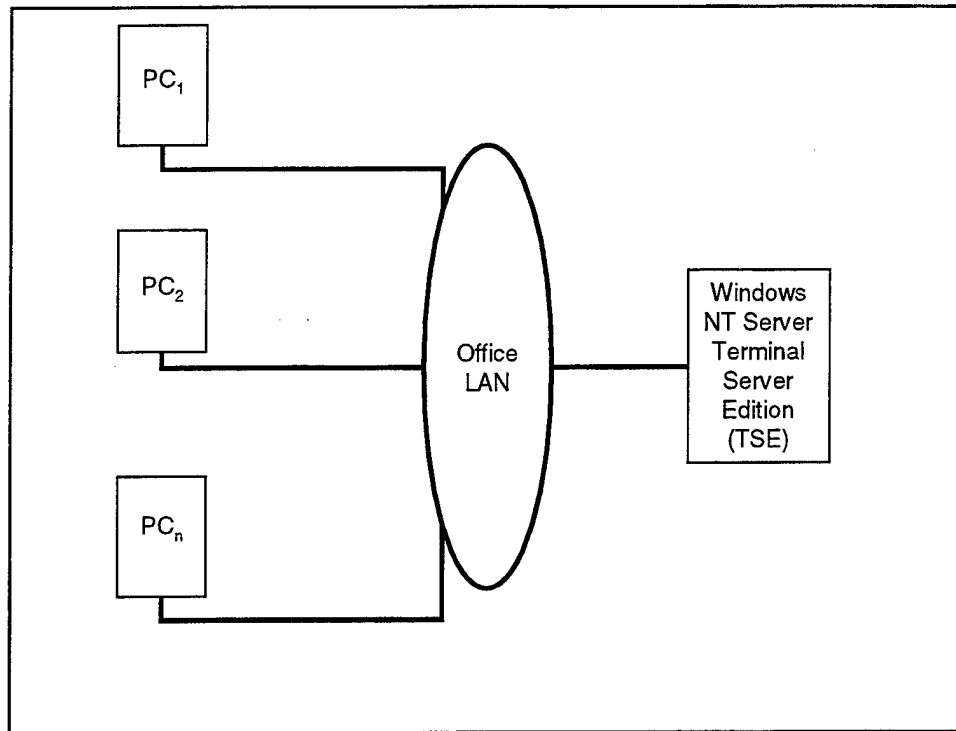


Figure 11. A Basic TSE LAN

The topology in Figure 11 is representative of a small TSE LAN. The TSE server provides operating system and application support to a group of users whose usage requirements are low to moderate with respect to network activity and CPU usage.

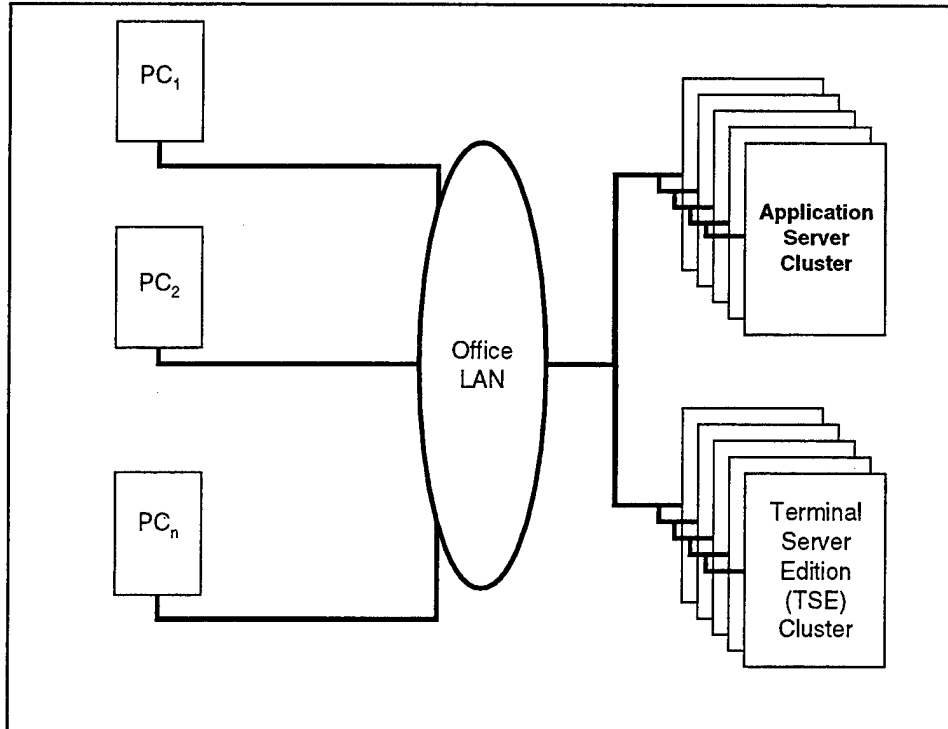


Figure 12. An Enterprise-Level TSE LAN

Figure 12 represents the type of LAN that uses TSE to serve many users or a group of users who have very high usage requirements with respect to network activity and CPU usage. There are many TSE servers and application servers working together to provide adequate support for such a LAN. The presence of an application server balances the load for more processor-intensive applications and reduces the stress on the processors for the TSE server cluster.

B. TERMINAL SERVER EDITION TOPOLOGIES FOR THE MLS LAN

The principle problem addressed in this chapter is how to deliver the operating system to the client with the greatest degree of integrity and the highest assurance of object reuse. There must be high assurance of security policy enforcement within the LAN. As mentioned above, TSE is capable of delivering the NT operating system to a variety of PCs including diskless network PCs and thin clients. The question addressed in the following sections is "Can any configuration of the MLS LAN with TSE be secure?" Figure 1 in Chapter I is

representative of a MLS LAN. It consists of a high assurance server (HAS) running an IMAP e-mail server as an example of the many shared resources and services desired on modern local area networks. Any number of clients running the Windows NT operating system are connected to the LAN. The TCBE in each of the clients allow the formation of a trusted path and secure communications between the client and the HAS while ensuring that object reuse is maintained at the client. The rest of the configurations considered in this section will build upon Figure 1 using the TSE server to deliver the operating system to the client. The clients in TSE-based networks that already run their own operating system loaded from local hard-drives suffer the same security weaknesses of networks described in Chapter III. In this chapter, the client will consist of diskless network PCs or thin clients.

1. Case 1 – TSE on the LAN

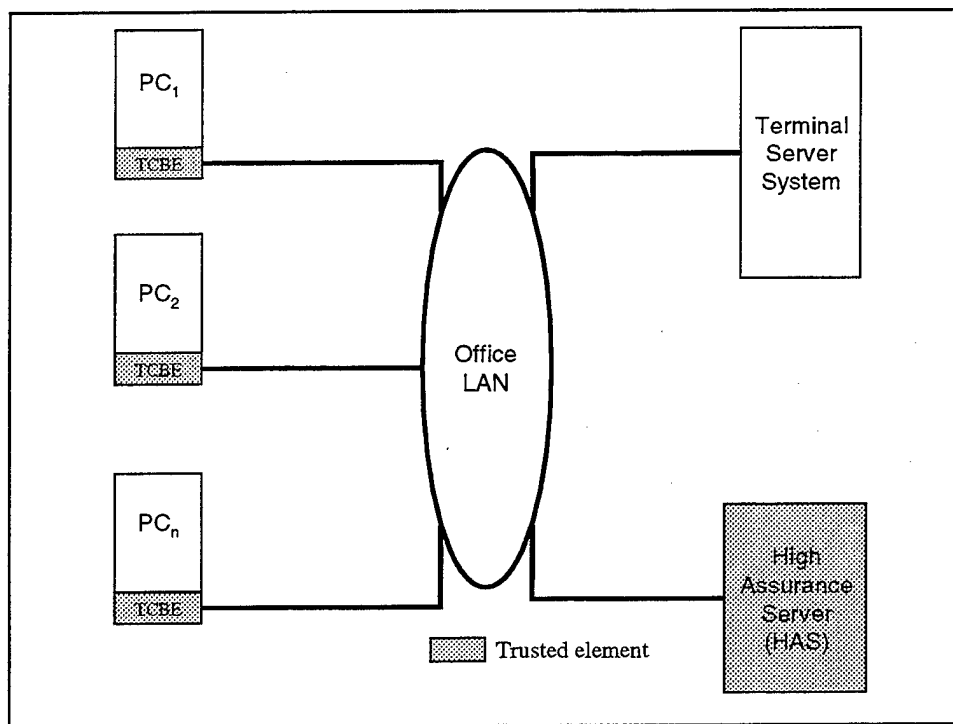


Figure 13. TSE Server as a Peer on the MLS LAN

The topology in Figure 13 represents the simplest addition of the TSE server to the MLS LAN. The HAS and the client/TCBE work as in Figure 1. The TSE server is added to deliver the operating system to the client.

a. How it Works

This network is made up of any number of PCs configured to communicate with a Terminal Server Edition server. Each PC is enhanced with a TCBE to facilitate the trusted path, and encrypted communications with the HAS. The user would turn on the PC, if it is off, and invoke the secure attention key (SAK) to begin logon and session negotiation with the HAS. Once Identification and Authentication are complete with the HAS and the session level has been successfully negotiated, the operating system would be allowed to load (as the TCBE is controlling the bootstrap of the client PC) and the user would logon to the TSE. The TSE would then process all keyboard/mouse actions at the client. The client could run an e-mail client and access the IMAP e-mail services running on the HAS. The IMAP server would allow the user to access mail at classifications dominated by the session level negotiated with the HAS.

This configuration is very efficient and scales well. The number of users on a single server could range from fifteen to forty-five dependent on the usage patterns. If the expected number of PCs is large, the TSE would expand to a cluster of single or multiple processor TSE servers running load balancing software to optimize efficiency. The TSE on the LAN allows for the TSE to process the bulk of the network traffic without the interference of the HAS. This is better because the HAS is involved only with trusted path and protocol services such as e-mail services.

b. Security Analysis

The TCBE and HAS exchange two keys when the SAK is pressed and a session is negotiated. The first key is used for trusted path communications. The second is used for session communications. During the trusted path communications the client is in direct contact with the HAS and the TSE is not utilized. Subsequent communications in the

network require the TSE. Since the TSE has no knowledge of the encryption algorithm and keys used between the HAS and the client/TCBE, communications on the network must be in the clear. Use of TSE compatible public key algorithms would be imprudent as Microsoft implementations of these protocols have been known to contain documented vulnerabilities, although these vulnerabilities have since been patched. [Ref. 21]

There are numerous open vulnerabilities to confidentiality on this LAN. The first takes advantage of the open communications between the client and the TSE. Any user attached to the LAN can eavesdrop on LAN communications and see all the data accessed. Thus, if a user is logged on and accessing data at "HIGH" and another user is logged on at "LOW", the user at "LOW" can see all the communications and information is leaked.

The second vulnerability is in the TSE itself. TSE uses discretionary access controls (DAC). Such systems are vulnerable to malicious code. Each user logged onto the TSE is a process running in its own memory space. Applications started on behalf of the user act with the user's privileges. A user with "HIGH" session level might have a Trojan Horse running that writes some place that a user at "LOW" can access. Some processes can be written to operate in the executive mode and thus have greater privilege. Such processes can acquire unrestrained access to all memory in the system. Such processes may grab information from another's memory space and copy it into its memory space for exfiltration and again information is leaked. Finally, all users must log onto the TSE for access to NT applications. The single TSE caches data for its own efficiency and information could be found in swap files maintained on permanent media. The TSE's capabilities to adequately separate user processes is unproven and does not meet the needs for security in the context of the MLS LAN. It could be assumed that there is open mixing of information in such a system requiring that there be some physical means for separating users of different classification levels.

2. Case 2 – TSE Under the HAS

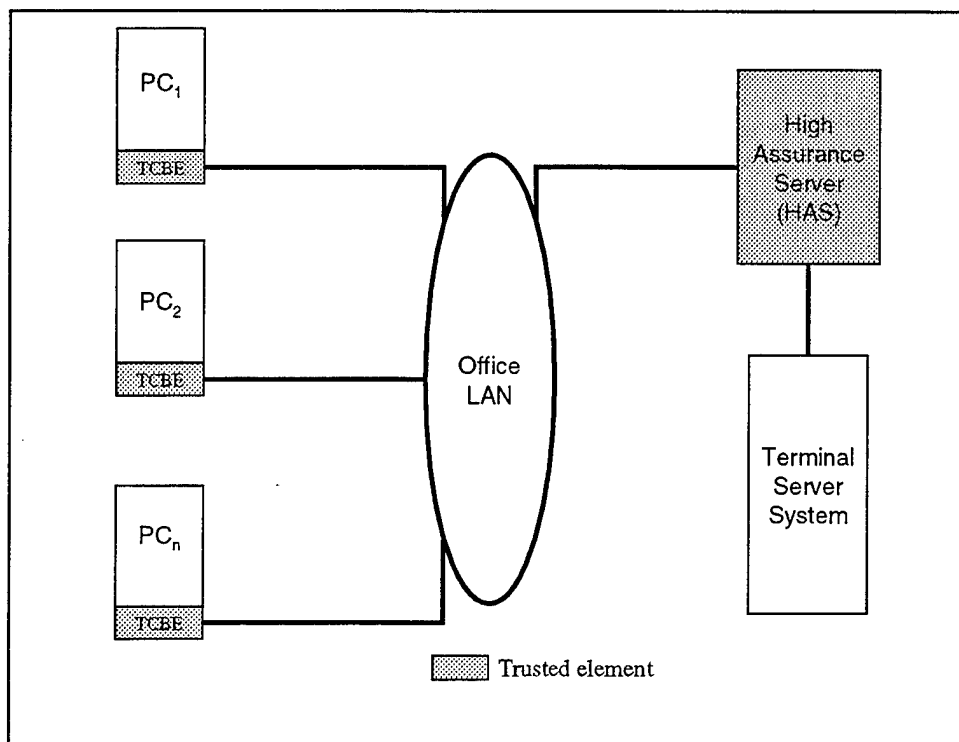


Figure 14. TSE and HAS in Series

Figure 14 represents a possible configuration where the TSE is attached to the HAS vice on the LAN (in parallel). All communications between the client and the TSE are mediated by the HAS.

a. How it Works

This configuration, though subtly different, provides a significant difference in performance and efficiency of the network, and some improvement in security. The client and the HAS are attached as in Case 1, so the trusted path and session negotiations are as before. The significant difference seen here is that the clients are not in direct communication with the TSE. This allows all communications on the LAN to be encrypted with the keys exchanged during the logon with the TCBE and the HAS, an improvement in security. The HAS and TCBE handle all key exchange, encryption and decryption. When the HAS-to-TCBE Identification and Authentication are complete, the client will be allowed to complete bootstrap

procedures and begin communications with the TSE. This means that the path to the TSE through the HAS must be enabled. The HAS could complete the I&A to the TSE on the user's behalf, or the user could receive the TSE I&A communications as a separate event in the logon process. The bulk of the traffic on the network would be the communications between the TSE and the client PCs for processing keyboard/mouse events, and graphical display objects. This must be encrypted/decrypted by the HAS and passed on to the TSE. If the user wishes to access e-mail, he/she must first activate the e-mail client in the TSE. This communications path is: client/TCBE to HAS to TSE. The application is activated on the TSE and makes a request to the HAS on behalf of the user. This communications path is: TSE to HAS. The HAS passes the information to the TSE which is then displayed to the client. This path is: HAS to TSE to HAS to client/TCBE. This chain of events unnecessarily burdens the HAS with processing all data to and from the TSE and client, however, it does allow privacy and security for data in transit. The HAS could be a definite bottleneck in LAN communications and must be modified to optimize traffic handling more efficiently.

This configuration does not scale very effectively. To handle a greater number of users and the expected traffic, the HAS as well as the TSE must be multiplied. This is too expensive and impractical.

b. Security Analysis

Security is improved in this configuration as encryption is allowed on the LAN for all traffic between the HAS-TSE and the TCBE/client. Data in transit on the LAN are not susceptible to eavesdropping as each access class and possibly user, is given a separate set of keys for communications. A hostile user on the LAN would have to gain access to the keys assigned to a particular user session and/or attack the protocol. There is no improvement in the security of the TSE itself. The TSE is still incapable of keeping data separate between classifications with any assurance, and is still vulnerable to malicious code running in its application layer. Leakage could occur within the TSE from users at "HIGH" to users at "LOW". Numerous timing and storage channels could be conceived for this unevaluated platform.

3. Case 3 – TSE Enhanced with TCBE on the LAN

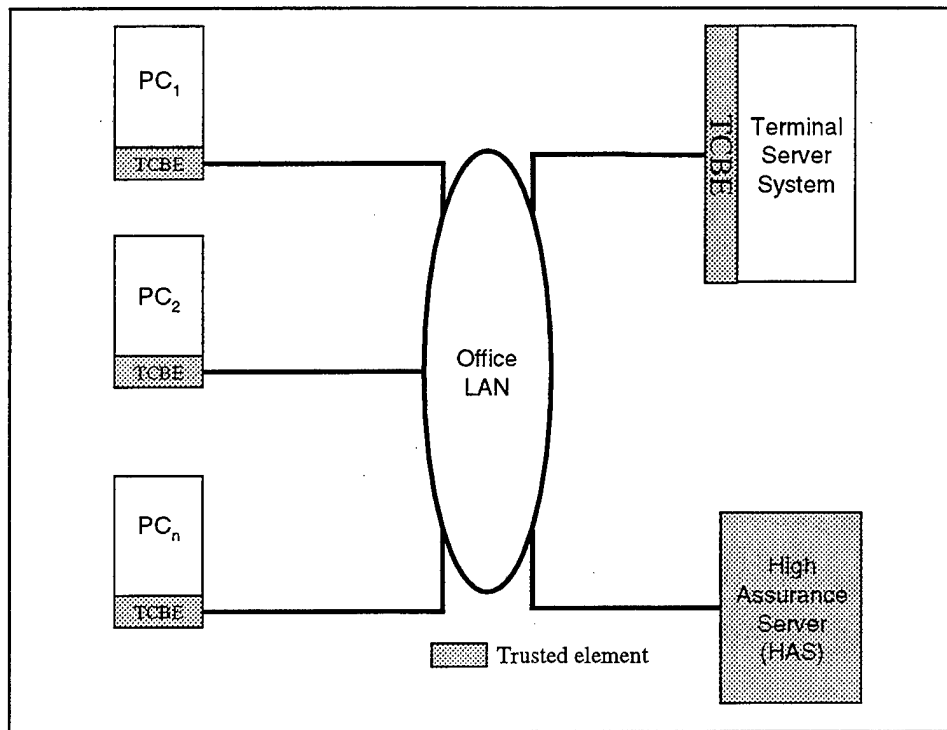


Figure 15. Enhanced TSE on the LAN

Figure 15 represents a MLS LAN with an enhanced TSE. The TSE operates as a peer with the HAS with respect to access to the LAN. The presence of the TCBE on the TSE allows the TSE to communicate securely with the client/TCBE and the HAS.

a. How it Works

This is very similar to Case 1, its topology is shown in Figure 15 and has the TSE on the LAN as a peer with the HAS. The obvious difference is the presence of a "TCBE" on the TSE system. This TCBE has similar functions to the TCBE placed on the client PCs. It is capable of handling key exchange with the HAS, computing cryptographic algorithms, and conducting a controlled bootstrap of the TSE host. It differs from the TSE's TCBE in that it must be able to manage keys declared for each client PC logged onto the TSE server where the TCBE on the client only has to manage its own.

A typical session would begin as in Case 1, where the user turns on the power and/or presses the SAK. The I&A is driven by the HAS, and when completed, the HAS may complete the I&A with the TSE on the users behalf giving the TSE's TCBE the appropriate keys for the session. Key exchange with the HAS, client, and TSE will be complex. There will be a separate trusted path key for each client and for the TSE shared with the HAS. This is required to provide logical separation for communications. Trusted path communications are always with the HAS alone. It is required to prevent the possibility of spoofing the trusted path to the client or the TSE. It also allows for each element to communicate privately, which is a requirement for the trusted channel. There will also be separate session keys shared among the clients, the TSE, and the HAS. The separate session keys provide logical separation and prevent capture of data on the line by other clients. The HAS and client must exchange two keys (one trusted path key and one session key), the HAS and TSE will have to exchange keys too (at least once with a trusted path key on the TSE's initial bootstrap, and one for each client that acquires a session key from the HAS). Since it is impractical to reboot the TSE for each change of session of each client because service will be interrupted for other clients, the trusted path key between the TSE and the HAS will need be re-negotiated in accordance to some key expiration schedule.

b. Security Analysis

This configuration allows more efficient runtime communication between the client and the TSE because it does not involve the HAS. It allows for encrypted communications on the LAN thus securing data in transit. This is a marked improvement to security from Case 1. A malicious user must have access to the keys for a target client PC in order to acquire the information transmitted between the TSE and the targeted client. Since keys are changed with each session, it is unlikely that the malicious individual will be able to use the data in his/her lifetime.

This configuration does nothing to secure the data as it is handled by the TSE itself. TSE still has the same vulnerabilities internally through data mixing in an untrusted and

insecure operating system. The only way to ensure that data are not leaking from “HIGH” to “LOW” in the TSE is to physically segregate the TSEs into discrete system high domains.

4. Case 4 – Multiple TSEs on LAN

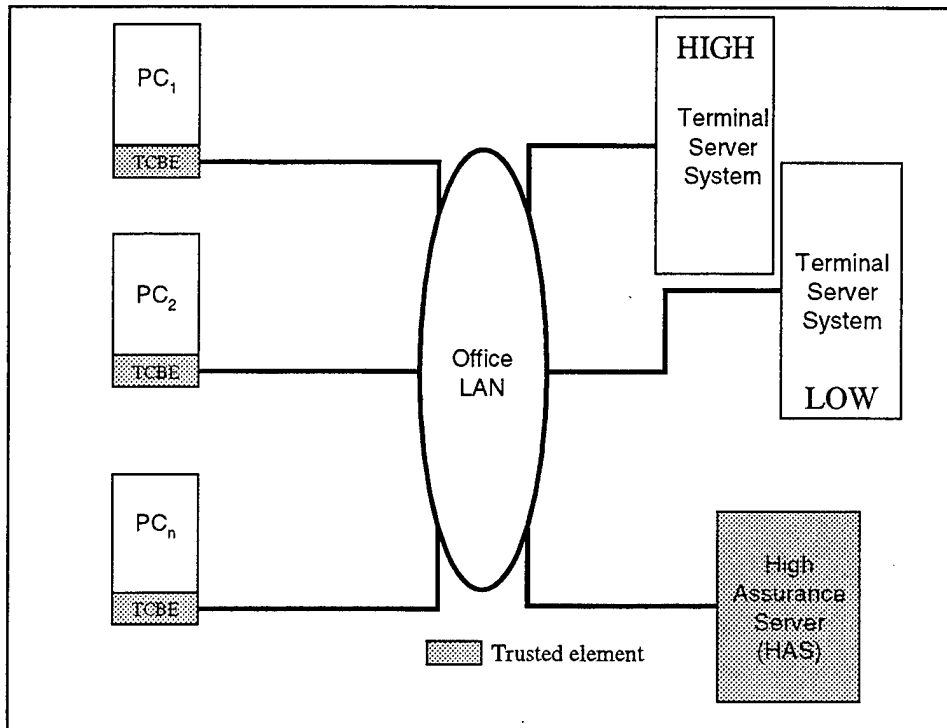


Figure 16. Multiple TSEs for Separate Classifications

Figure 16 depicts a MLS LAN with physically separated TSE domains for each classification level. Each TSE has access to the LAN for maximum throughput. Each TSE is allowed to access only one classification level to prevent mixing of information that may be stored by the operating system.

a. How it Works

This system builds upon Case 1 by physically separating the classifications that a TSE can handle into discrete system high domains. This means that a user who attempts to access data at level “HIGH” and below will access the data only through the TSE designated for “HIGH”. Similarly a user who attempts to access data at “LOW” will only access the data through the TSE designated “LOW”.

Looking at this in more detail, the user at the client PC invokes the system in a fashion similar to that of Case 1 by powering on the system and/or pressing the SAK. The HAS again drives the I&A for the user/client. When complete the HAS may complete the I&A with the appropriate TSE for the session level assigned. The user's PC will then complete the connection with that TSE and begin the session. Note that the TSEs are not enhanced with TCBEs as in Case 3 and so have no protected communications channel between: (a) the TSE and the Client/TCBE, and (b) the TSE and the HAS. So, in order for the TSE to communicate with the PCs and the HAS, data in transit must be in the clear.

b. Security Analysis

In spite of the physical separation between the TSEs, this configuration suffers the same problems as in Case 1: open attacks to data in transit by malicious listeners on the LAN, and exfiltration of data within the TSE itself. The fact that data on the LAN cannot be encrypted in order to allow the TSEs to function, means that any eavesdropper on the LAN can view all data that is moving there. A user logged in and accessing data from "HIGH" with the TSE designated for "HIGH" will receive her/his data on the same LAN and using the same protocol as a user at "LOW". It would be possible for such an eavesdropper who has legitimate access at "LOW" to gain access to the TSE at "HIGH" by spoofing the user who logged in at "HIGH". Using similar malicious code mentioned in Case 1, data may be leaked to this "LOW" user who probes the memory and swap space on the "HIGH" TSE.

A separate TSE is needed for each security level. This may be impractical when many access classes are required.

5. Case 5 – Multiple Enhanced TSEs on LAN

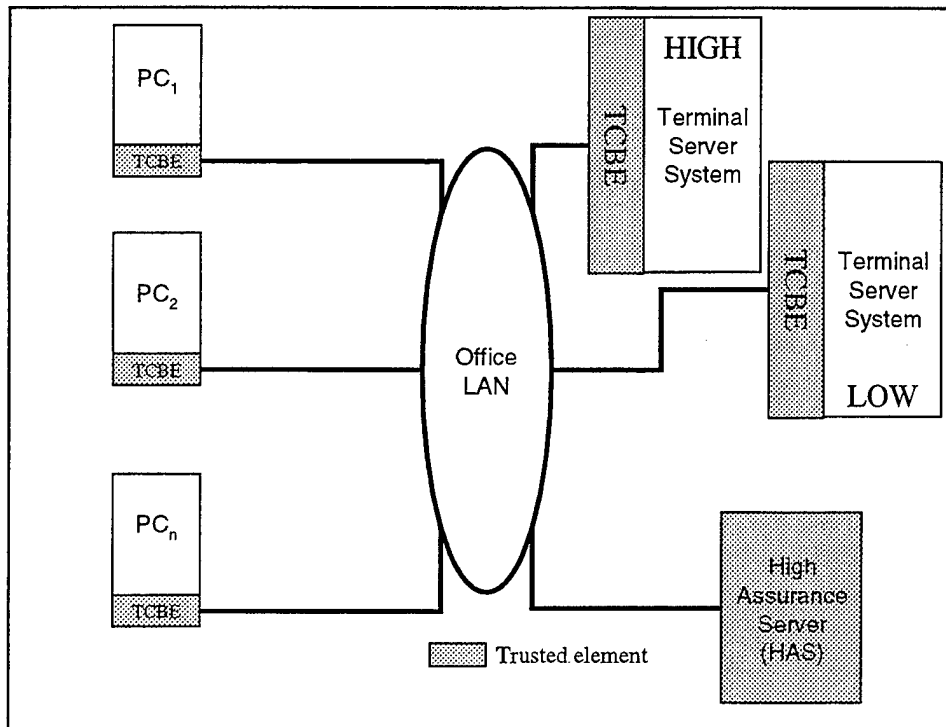


Figure 17. Enhanced TSEs at Different Classifications

Figure 17 is the more secure extension to Case 4 with TCBE enhancements applied to the TSEs. The presence of the TCBE on each TSE allows the TSEs to communicate securely with the client/TCBE and the HAS.

a. How it Works

The user who wishes access at "HIGH" turns on the PC and/or invokes the SAK to initiate I&A with the HAS. This results in the generation of trusted path and session keys for that client. Similarly a user accessing "LOW" will receive different trusted path and session keys for access at "LOW". This creates a complete logical separation of data in transit securing this information. The HAS could then contact the separate TSEs at "HIGH" and "LOW" on behalf of the users at "HIGH" and "LOW" respectively. There would be a unique key established for each of the TSEs for trusted path communications with the HAS. The HAS would exchange session keys with the TSE at the appropriate level on behalf of the users,

and complete the I&A for the user. In this way, the user at "LOW" will have access only to his/her own data through the TSE at "LOW" only (similarly for the user at "HIGH".)

b. Security Analysis

This configuration maximizes security of data in each component and data in transit for a MLS LAN. Communications on the LAN are encrypted for trusted path and session traffic by separate keys. The HAS drives I&A for the PC/TCBE and TSE/TCBE.

This prevents eavesdropper from gaining information from traffic on the LAN. This encryption of traffic also enforces the separation of the TSEs. It would be virtually impossible for the user at "LOW" to spoof the "HIGH" TSE as keys do not persist past a single session for a particular user.

No amount of added encryption can prevent internal attacks on the TSE, but proper design of the TCBE can prevent the exfiltration of information from "HIGH"-to-"LOW". The TCBE and its encryption algorithms must always be invoked when transmitting from either the client or the TSE. By preventing the bypass of encryption when transmitting data, the user at "LOW" can never see "HIGH" data.

An attacker could get malicious code into a TSE and probe information accessed by another user. This attacker would first need legitimate access (i.e. the appropriate clearance level) to the TSE at the level being targeted. Any loss of information by such an attack would then be kept at that level, and information is not leaked from "HIGH"-to-"LOW", but rather movement of information from "HIGH"-to-"HIGH" would only constitute a possible violation of the a discretionary requirement for access to information.

A separate TSE is needed for each security level. This may be impractical when many access classes are required.

6. Case 6 – An Ideal Solution

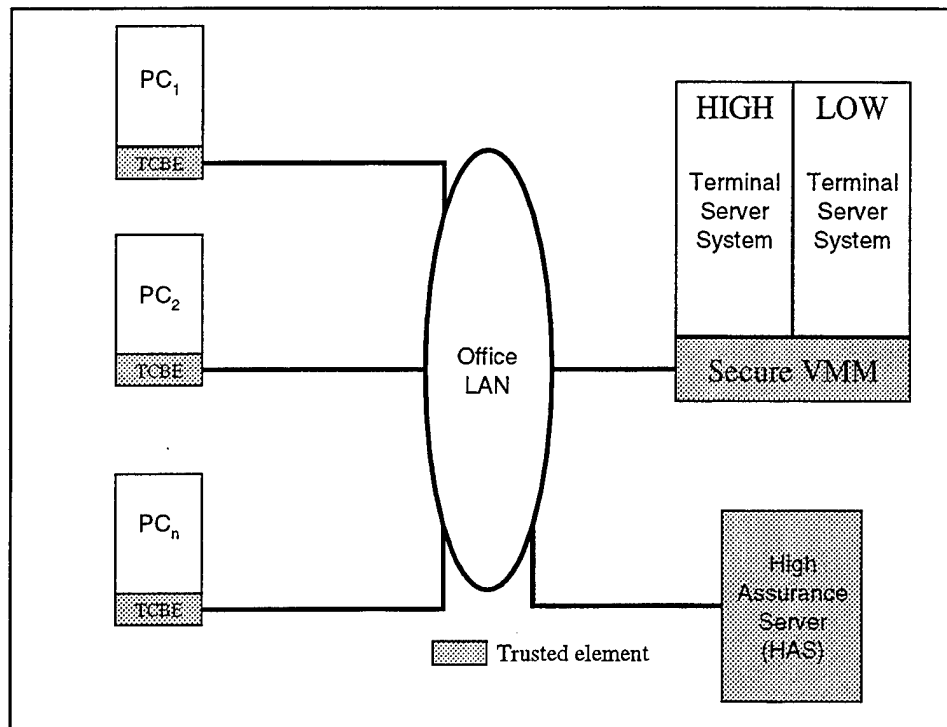


Figure 18. TSEs Running on a Secure Virtual Machine Monitor

The topology in Figure 18 represents a theoretical LAN that utilizes a secure virtual machine monitor to keep user instances of the NT work environment logically separate.

a. How it Works

A virtual machine monitor (VMM) is capable of creating and controlling virtual replicas of a particular processor and its operating environment including address space for RAM and any permanent media required. The Intel 80286 and 80386 would operate in a mode that virtualized 8086 processors. Research is underway to determine if the Intel Pentium processor is virtualizable. [Ref. 22] A VMM would be able to create a separate environment for multiple operating systems that would be completely separate from each other. In a secure VMM, measures are taken to provide assurance that the separate environments are secure from attack by adjacent environments and the operating systems/applications running there. The heart of what TSE does is to present to multiple users, the interface and functionality of

the Windows NT operating system. The VMM would then provide a virtual machine for TSE to run on. For each separate classification level, a separate TSE would be dynamically created each capable of supporting multiple users. TSE is not a VMM. It virtualizes users as processes. Each user shares the TSE environment and the applications that TSE is capable of running.

In this topology, the secure VMM creates virtual machines for an operating system that is capable of running Windows NT applications and sending the Windows NT work environment to diskless workstations as clients, in this case TSE. For each user who logs onto the LAN, the HAS will control I&A and then commission the secure VMM to create or invoke a TSE to service that user's session level. The TSE will, in turn, create an environment for the user on the appropriate TSE for the classification level negotiated. If no TSE server exists at the appropriate level, the VMM will start one. The TSE server, at the correct level, will then handle delivery of the working environment to the client. Trusted path and session key exchange between the HAS and each client/TCBE could occur with separate keys for each client/TCBE. This is necessary to allow private communications without the threat of other users eavesdropping. The HAS will have to exchange keys with the VMM for the VMM's trusted path communications. The HAS will also have to exchange session keys with the VMM for each client/TCBE logged on. The secure VMM would carry on the responsibilities of the TCBE for the TSE in Cases 3 and 5 with respect to those keys.

b. Security Analysis

Security is maintained in the three trusted elements in the LAN. The HAS maintains separation of data, the secure VMM maintains separation of data in use by the operating system and the applications running on the user's behalf, and the combination of all three maintains security of data in transit on the LAN between the secure VMM, the HAS and the client/TCBE. Separation and security of data rests on the logical separation provided by cryptography and the established capabilities of the Wang XTS-300. A secure VMM has not been developed for the Intel architecture or for the Windows NT operating system, so this solution is currently only hypothetical.

C. TSE GENERAL ANALYSIS AND CONCLUSIONS

All configurations with TSE must contend with the untrustworthiness and insecurity of TSE itself. Since information handled by TSE does not have labels, TSE servers must be physically or logically separated and classified to the highest level of the material that would be handled by them (referred to as "system high" domains.) This separation must be buttressed by encryption and protocols to ensure that connections to "HIGH" classified TSE servers cannot be spoofed by users cleared only for "LOW". Data in transit must also be encrypted to prevent the hostile user on the LAN from collecting information as it moves from the TSE to the client, or the HAS to the TSE. From this analysis, Case 5 in Figure 17 is the only candidate topology with adequate security.

MLS LANs must have at least as many TSE servers as there are classifications accessible on the LAN. This is a difficult scaling problem with respect to classifications and the number of users who access services on the LAN. If a new classified data source is added to the LAN, a new TSE server must be configured to support it or it must be accessed only by TSE servers whose classifications dominate the new level in a read only status. If certain levels have heavy access (UNCLASS in a traditional network), the workload may exceed a single TSE server's capacity. This creates a different scaling problem, as clustered servers must be brought on line with load balancing software that might complicate security issues. This problem might be mitigated somewhat by the fact that the TSE capabilities are nearly linear with the number of processors used. This means that a quad-Pentium server can handle four times as many users as a single-Pentium server. If application access causes congestion on the network through high network activity or processor demand, a TSE Cluster/Application Server Cluster configuration may be made necessary. This configuration would be similar to Figure 12 but with trusted enhancements to the TSE cluster, Application Server Cluster, and the addition of the HAS for protocol access.

Except for Case 6 – which is completely theoretical due to the absence of a secure VMM, TSE does not provide a suitable solution for delivering a Windows NT GUI and access to applications that may be run on Windows NT in a MLS LAN.

THIS PAGE INTENTIONALLY LEFT BLANK.

V. CONCLUSIONS AND FUTURE WORK

A. RELATED WORK

1. Novell Trusted Workstation Partnership

Attempts to develop multilevel secure local area networks have been fraught with the difficulty of providing a user-satisfying interface at a low cost. There have been very few recent commercial attempts in this area of research. One such commercial strategy for the development of trusted workstations was the Novel Trusted Workstation Partnership. [Ref. 23] In this project, Novell was working to expand the number of clients to use with its Class C2 evaluated network software, NetWare 4.11. In their partnership document, Novell was attempting to address the problem of too few secure networking components available in industry. They defined an open Netware Global Security Architecture (NGSA), and used the Trusted Network Interpretation (TNI) of the Trusted Computer Systems Evaluation Criteria (TCSEC) to form a Network Security Architecture and Design (NSAD) document. This document would be use by developers of networking components to build secure network devices to use in this architecture. This network architecture was intended for Class C2 evaluation which is insufficient for multilevel secure application but the analysis provided for developing secure clients was useful.

Only one product was identified by the work. A company called SISTex produced a workstation client that successfully partitioned a user domain and a TCB domain. It was designed as an add-in board for a client workstation. There were some difficulties with the architecture.

- The client was designed as a diskless workstation. This wouldn't work for the selected version of Windows NT. Windows NT Terminal Server Edition could provide the NT work environment to a diskless workstation, but this is unsuitable because of the inherent problems with TSE that are identified in this thesis.

- The add-in card did not control other I/O devices for the client workstation. This could allow uncontrolled information flow.
- The device was designed for the ISA bus, which has inferior data transfer rates to the PCI bus, and would be insufficient for current high-speed Ethernet networks and high capacity high data-rate hard drives needed in today's PCs.
- The principle software feature on the add-in card was a file server, which was only a small function of our perceived TCBE. Additional needs included support for control of independent peripherals such as keyboard, and display. Also needed is software to support secure connections, and Identification and Authentication.
- The cost of the add-in card was excessive (Novell's analysis).
- The company no longer exists, and the product is unavailable.
- There appears to be no documentation available for the work outside this broad overview of features and considerations toward architecture.

2. Media Encryption Management System (MEMS) [Ref. 24]

Another body of work that provided insight into the work of this project was the Media Encryption Management System (MEMS). This product encrypted hard drives and conventional floppy drives by interposing hardware and software between the operating system and the storage media. It is intended to provide an operating system independent approach to media encryption. All information written to the storage media is encrypted, and all information read from the storage media are decrypted. There are two components to the MEMS architecture, a cryptographic peripheral that performs all cryptographic functions and the control software that directs all data flow to the cryptographic peripheral.

The control software handles user validation before unlocking the storage media. The control software acts as a reference monitor for drive accesses in that it is designed to capture all accesses to the drive and it is always invoked. The element of the control software that captures all drive accesses is called the Intercepting Device Driver. It is designed to monitor operating system activity and intercept all accesses to the media. All reads and writes are handed off to the software that manages the cryptography.

The analysis provided on this system gave insight into the problems of controlling the access to devices. [Ref. 25] Direct access to devices attached to PCs may be accomplished through bypassing the operating system and sending instructions directly to the BIOS addresses for the devices. MEMS is designed to intercept operating system accesses to devices that are a level of abstraction above BIOS calls. If the operating system is bypassed by a program, it directly accesses the BIOS routine for devices, and MEMS is unable to intercept these calls. This type of access is common in DOS applications, as it is much faster. It also occurs in the Windows operating system with direct access to the swap file that is used as secondary (virtual) memory.

The analysis also described situations where direct access to the memory address that controls the device not only bypasses the operating system, but also bypasses BIOS calls. This occurs commonly in access to I/O ports such as serial ports. [Ref. 25, p. 1] It is also done with hard drive access on systems that are capable of exploiting ultra-fast data transfer capabilities of current motherboard chipsets and hard drives. [Ref. 14]

From this research, we must acknowledge the powerful capabilities of the operating system, BIOS, and direct access to devices on PCs that must be controlled. We must be able to demonstrate that all accesses are monitored, and that our control over the device is complete.

B. RECOMMENDATIONS FOR FUTURE RESEARCH

1. Trusted Paths/Channels Into Network Clients

The trusted path/channel is the most important link between the TCBE and the TCB. It must be an unspoofable, secure connection between the TCBE and the XTS-300 TCB that provides high integrity identification of each end of the channel. It is required in order to access the necessary functions in the XTS-300 TCB from the TCBE. Research is needed on how to signal securely that a trusted path connection is desired, and once that intent is made known how to establish the connection. Once a trusted path is established, there is the problem

of how to maintain it with security. For our project, the necessary protocols to exchange cryptographic keys, support for various cryptographic algorithms, unique ways to identify user and hardware, and extensions to secure connection databases kept on the XTS-300 are also included.

2. Disk Ghosting Technology

Symantec's Ghost software is designed for a DOS environment. It is capable of making exact duplicates of information on hard disks preserving partition structures and file order. It is also capable of storing images of a drive in remote locations for use in restoring drive configurations remotely. Its highly configurable graphical interface and command line input make it a valuable tool in accomplishing the higher integrity delivery of an operating system and applications to client workstations. It is also invaluable for enforcing total object reuse from session to session with respect to the client's permanent media. As DOS is not the preferred operating system for the TCBE, research into creating similar capabilities in a compact highly configurable and open source operating systems is necessary. Further investigation into the appropriateness of the ghosting technology as an operating system delivery method, may be sought.

3. Object Reuse in Commercial Personal Computers

This thesis presents ways to better enforce object reuse in the client PC's permanent media. There are numerous storage objects on today's commercial PC's. Storage objects include any device that has the capability of storing bits of information. Such storage objects include RAM, buffers, registers, programmable controllers, and perhaps some CMOS and FLASH ROM used for system or expansion BIOS. Each of these must be located, their interface must be determined, and ways to overwrite them between sessions must be discovered.

4. BIOS Extensions in Trusted PCI Components

The TCBE will require highly efficient software to provide a secure and appropriate interface to the host PC. Typical add-in cards accomplish this through the use of expansion BIOS. Research into the creation of such an expansion BIOS for the TCBE must be completed to establish a secure means of communicating with the host PC. Further research into the limit of control that can be established on a host PC by an add-in card through the use of BIOS extensions is also needed. Controlling pathways that utilize the expansion BIOS while providing assurance, integrity, and security could reduce time to prototype considerably.

5. Trusted Hard Drive Encryption

The use of encryption to provide logical separation of information stored on hard drives within a network client is necessary. Efficient and secure means of encrypting permanent media must be discovered. Hard disks today are extremely fast. As fast as they are, they are the slowest component on a PC and often limit the PC's ability to function at peak potential. Encrypting the permanent media places an even slower restriction on the hard disks' capabilities. Research exploring the effect of encryption on performance is needed. Furthermore, discovering ways to provide sufficiently strong encryption for the information being handled while minimizing its effects on system performance is needed.

6. Key Exchange Protocols

Cryptography will play a key role in any networked system that needs confidentiality and integrity in its transmissions. With public key cryptography or cryptography where there are shared secrets, keys need to be exchanged. Such key exchange needs secure protocols to ensure that the keys are not compromised. Research into the appropriate protocols needed for the various key exchange scenarios within the NPS MLS LAN is needed. There will be logically separate communications paths for trusted path and session communications. Under some of the operating system delivery methods, there are client- and/or user-specific key exchanges associated with permanent media encryption. This research is closely tied to research on the trusted path.

7. Identification and Authentication (I&A)

By modifying the XTS-300 I&A module, user identification and authentication processes could be enriched to include token based I&A. This so-called "two-part" I&A includes the information stored in something that the user has physical security over such as a smart card with some certificate and/or key information, and something that the user knows like a password or a personal identification number (PIN).

8. Secure Bootstrap

Setting a secure foundation for an operating system and the peripherals that are installed into a client PC is something that has not gotten a lot of attention. William Arbaugh introduced secure Bootstrap in his doctoral work and in his work with AEGIS. [Ref. 10] The PC bootstrap process includes integrity checks of software and firmware as it transitions from one stage to the next. These checks continue all the way to operating system load. Secure Bootstrap could be implemented on the client in the MLS LAN to provide greater assurance of the integrity of the client PC.

C. CONCLUSIONS

The Naval Postgraduate School Multilevel Secure Local Area Network (MLS LAN) is intended to provide true multilevel access to information over a LAN. This project relies on the Trusted Computing Base of the Wang XTS-300, a Class B3 evaluated computing device, to maintain the multiple classification levels that could be accessed by the LAN separated in accordance with the policy entered on the machine. The purpose of the Trusted Computing Base Extension (TCBE) is to provide a secure connection to the XTS-300 acting as a server in the LAN, provide access to the TCB on the XTS-300 for Identification and Authentication purposes, and to provide necessary security features on the client PC to properly constrain the untrusted commercial operating systems and applications that are used to access the information made available by the XTS-300 and the protocols that will be supported on that system. Specifically the NPS MLS LAN prototype will provide IMAP e-mail services to the

client PC's. The user will be able to access the e-mail in his/her account by using commercial e-mail browsers such as Outlook, Outlook Express, Netscape Mail, or Eudora. All of these programs provide the type of user-friendly interface and functionality that users in office environments desire to maximize productivity.

This thesis identified functional interfaces required on the TCBE to be able to adequately secure the PC and to ensure secure communications with the high assurance server. Suggested methods for implementing these interfaces were presented. These implementation choices were analyzed to identify security risks and benefits.

We also identified possible ways for securely delivering a commercial operating system to the client where integrity and confidentiality are assured. These methods sought to maximize object reuse control over the storage objects found in the permanent media on which the operating system must reside. Advantages and disadvantages of each delivery method were presented with emphasis on the security provided to the client and the information secured on the high assurance server.

Finally, an analysis of the suitability of Windows NT Server 4.0 Terminals Server Edition for use in the NPS MLS LAN was conducted. In this analysis various topologies are discussed. An analysis of the security risks and benefits for each topography are presented.

We have reinforced the feasibility of the NPS MLS LAN through an improved knowledge base of the requirements for the TCBE. Many of the TCBE's interfaces and their security risks/benefits and better methods of operating system delivery that enhance object reuse and improve integrity and secrecy at the client have been identified. With this, a greater knowledge of the requirements and capabilities of Windows NT as a client operating system has been obtained, and we have come closer to a viable proof of concept for the Multilevel Secure LAN at NPS.

THIS PAGE INTENTIONALLY LEFT BLANK.

APPENDIX A. INTEL 440BX MOTHERBOARD

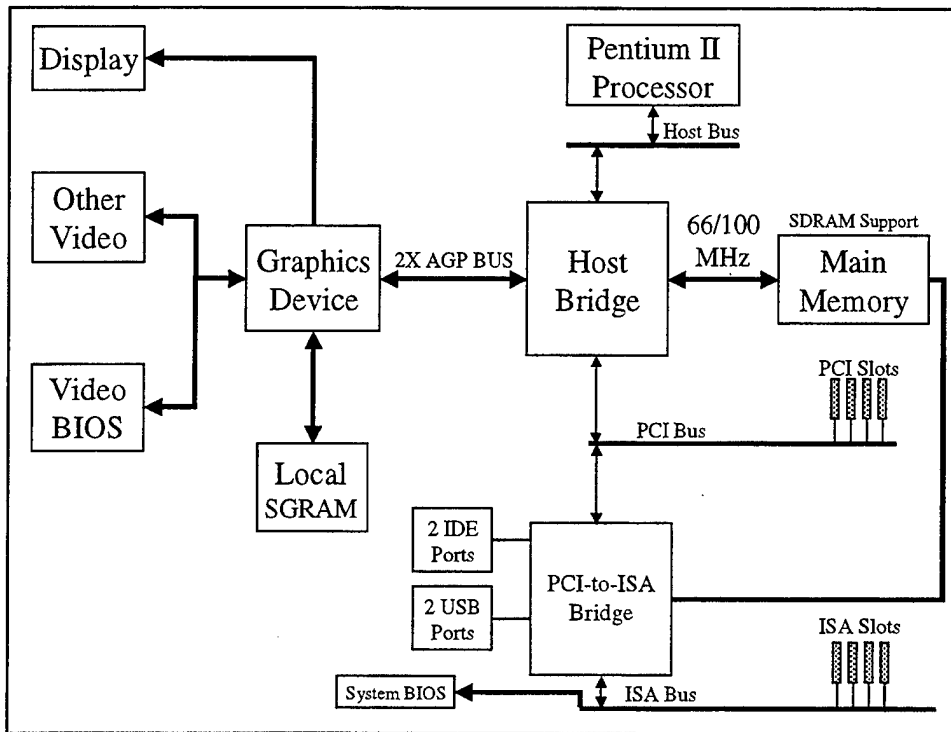


Figure 19. A Basic Block Diagram of an Intel 440BX Motherboard

Figure 19 represents a typical motherboard design. The diagram shows the major components that make a PC work. Of particular interest to this project are the capabilities of the PCI bus and the supporting chipset for this class of motherboard.

A. THE PROCESSOR

The CPU in this case is a Pentium II Processor capable of decoding three instructions per clock cycle and operates at speeds up to 450 MHz. The CPU is connected to the host bus via a Slot 1 single edge connector. Slot 1 is a connector not unlike PCI and ISA slots, but with 242 pins. This configuration takes up much less room than a flat pin package and provides the support for the large heat sink and/or fan that is required to prolong CPU life.

B. BUSES

The host bus can run at 66 MHz or 100 MHz dependent on the supporting chipset. The host bus is connected to the rest of the PC by two bridges. The primary or host bridge is implemented in the PCI/AGP controller (PAC). It controls the PCI and AGP buses.

The AGP bus runs at 66/133 MHz making data transfers as high as 500 MB/s (Megabytes per second). It is dedicated to the display/rendering of video images. It is capable of addressing its own memory pool, and when necessary address and move data to/from main memory through the PAC. To accomplish the latter, it utilizes a special addressing scheme, called pipelining, where the main memory requested must be sequential. Moving memory across the data lines on both the rising and trailing edges of the clock pulse doubles the throughput. This memory may be written and accessed much faster than random access because data transfers only require a start address and size. This allows the data to be streamed very rapidly.

The PCI bus can operate at 33 or 66 MHz. The PCI bus is capable of functioning without direct supervision of the CPU. This allows it to support a greater variety of devices with platform independent development.

- Peak bandwidth is four-times higher than the PCI bus thanks to pipelining, sideband addressing, and data transfers that occur on both rising and falling edges of the clock.
- Direct execution of texture maps from system memory. AGP enables high-speed direct access to system memory by the graphics controller, rather than forcing it to pre-load the texture data into local video memory.
- Less PCI bus congestion. The PCI bus attaches a wide variety of I/O devices; such as disk controllers, LAN chips, and video capture systems. AGP operates concurrently with, and independent from, most transactions on PCI. Further, CPU accesses to system memory can proceed concurrently with AGP memory reads by the graphics controller.

- Improved system concurrency for balanced PC performance. The Pentium II processor can perform other activities while the graphics chip is accessing texture data in system memory.

C. BIOS BASICS

BIOS stands for Basic Input/Output System. It is responsible for many functions at the hardware/firmware level in the PC. The BIOS is responsible for bootstrapping the operating system and communications between programs and the components that make up or are added to the PC. The BIOS accomplishes this through a group of assembly language routines. The BIOS works with the chipsets, such as Host-PCI-AGP and the PCI-ISA/IDE bridges, that make-up the motherboard and control access to data buses, system memory, and graphical display engines.

BIOS provides facilities for power-on-self-test (POST), bootstrap loader, and hardware initialization. POST performs checks to make sure that the motherboard is working by exercising components accessible from the motherboard. This check is cursory and does not check to see how well the equipment is working, just that it responds. The bootstrap loader locates and loads the operating system. This could be on a CD-ROM, floppy, hard drive, or other removable media. The location is determined by the CMOS data that is used by the BIOS. Each device must be initialized when power is applied. This is accomplished by the hardware initialization routines. Initialization could include setting registers on the device, setting the base memory address which is the buffer that exists in some expansion cards, or setting the base I/O address which is the conduit for message passing between the device and the CPU or the application that gets access to the device. [Ref. 14]

BIOS code is most frequently written to flash ROM. This is a programmable read-only non-volatile RAM that is similar to EEPROMs or electrically erasable programmable read-only memory. The name for flash ROM seems to be contradictory. How can something be read-only and programmable? The answer is a jumper on the motherboard that changes the state of a pin on the flash ROM chip and the proper access sequence sent to the chipset that controls

access to the BIOS. These allow the flash ROM to be written to. For this reason, the security of BIOS stored in flash ROM is suspect. Since supplying a twelve-volt signal to the device erases the chip or makes it writable, it is feasible to modify the BIOS. It is much harder to rewrite the BIOS without having physical access to the motherboard to set the proper voltage to the write pin of the flash ROM. Unfortunately, many motherboard manufacturers install and ship their motherboards with this jumper set to write. A denial of service attack, like the one accomplished by Chernobyl, exploits this to rewrite the boot block of the BIOS and make the PC unbootable. [Ref. 26]

APPENDIX B. SYMANTEC GHOST

Symantec Ghost is capable of cloning or making an identical copy, of disks or partitions. It can copy disk-to-disk or partition-to-partition. Ghost can also make an image file of a disk or a partition for transporting from machine to machine or for archival purposes. This image file can be used as a template for use on many disks. Some particular features of Ghost are:

- Ghost doesn't require the use of FDISK to partition a hard drive or FORMAT to prepare the partition for information to be copied onto a hard drive.
- The source and target disks can be different sizes. If they are the same size, Ghost copies uses sector-by-sector copy.
- Ghost supports FAT12, FAT16, FAT32, and NTFS file formats.
- Ghost can copy drives in the same computer or between two separate computers that are connected by Ethernet or parallel ports.
- A Ghost image file can be stored on a network server, CD-ROM, Superdisk, JAZ or ZIP drive, or other removable media.
- Ghost runs in DOS with a simple graphical interface. Ghost can also be automated through command-line switches.
- Save and load image files to and from a file server.
- Save and load image files to and from removable media.
- Clone multiple target PCs using multicasting.
- Ghost copies in-use files that other backups miss.

Ghost can automatically resize partitions if disk sizes are not the same. This is especially useful when moving from a smaller disk to a larger.

THIS PAGE INTENTIONALLY LEFT BLANK.

LIST OF REFERENCES

1. Brinkley, D., Schell, R. *What is There to Worry About? An Introduction to the Computer Security Problem*, Information Security: An Integrated Collection of Essays, First Edition, IEEE Computer Society Press, Los Alamitos, CA, 1995.
2. *Glossary of Computer Security Terms*, NCSC-TG-004 Version 1, National Computer Security Center, 21 October 1988.
3. *Department of Defense Trusted Computer System Evaluation Criteria*, DoD 5200.28-STD, National Computer Security Center, December 1985.
4. *Evaluation Criteria for Information Technology Security*, ISO/IEC JTC 1/SC27 N 2161, ISO/IEC 15408:1999 (E), 18 December 1998.
5. *Information Technology Standards Guidance*, Version 98-1.1, Department of the Navy DON CIO ITSG Integrated Product Team, 15 June 1998.
6. *Navy Virtual Intranet: Functional Architecture and Concept of Operations*, Draft Version, Naval Virtual Intranet Integrated Process Team, 11 December 1997.
7. Loscocco, P., Smalley, S., Muckelbauer, P., Taylor, R., Turner, J., Farrell, J., *The Inevitability of Failure: The Flawed Assumption of Security in Modern Computing Environments*, 21st National Information Systems Security Conference, Arlington VA, 8 October 1998.
8. *Trusted Network Interpretation of The Trusted Computer System Evaluation Criteria*, NCSC-TG-005 Version-1, National Computer Security Center, 31 July 1987.
9. *Evaluation Criteria for Information Technology Security – Part 1: Introduction and General Model*, ISO/IEC JTC 1/SC27 N 2161, ISO/IEC 15408-1: 1999 (E), 18 December 1998.
10. William A. Arbaugh, Faber, D., and Smith, J., "A Secure and Reliable Bootstrap Architecture", Proceedings of the 1997 IEEE Symposium on Security and Privacy, Oakland, CA, 1997.
11. Hackerson, J. *Design of a Trusted Computing Base Extension for Commercial Off-The-Shelf Workstations (TCBE)*, Naval Postgraduate School, Monterey, CA, September 1999.
12. Bryer-Joyner, S., Heller S. *Secure Local Area Network Services for a High-Assurance Multilevel Network*, Naval Postgraduate School, Monterey, CA, March 1999.

13. *Evaluation Criteria for Information Technology Security – Part 2: Security Functional Requirements*, ISO/IEC JTC 1/SC27 N 2161, ISO/IEC 15408-2: 1999 (E), 18 December 1998.
14. Croucher, P. *The BIOS Companion*, First Edition, Advice Press, December 1998.
15. Schneier, B., Shostack, A., *Breaking Up Is Hard To Do: Modeling Security Threats for Smart Cards*, First USENIX Symposium on Smart Cards, USENIX PRESS, 1999.
16. Smith, S., Weingart, S. *Building a High-Performance, Programmable Secure Coprocessor*, Computer Networks, Elsevier Science B.V., November 1998.
17. *Computer Security Subsystem Interpretation of the TCSEC*, NCSC-TG-009 Version 1, National Computer Security Center, 16 September 1988.
18. Space and Naval Warfare Systems Command; “Secure Windows NT Installation and Configuration Guide” Version 1.3, SPAWAR PMW-161 Report, December 1998.
19. *Start Here – Basics and Installation – Microsoft Windows NT Workstation Version 4.0*, Document Number 000-28620, Microsoft Corporation, 1996.
20. *Microsoft Windows NT Server 4.0 Terminal Server Edition Resource Guide*, Afinity Publication Division, Penn Well Publishing Company, Seattle WA, 1998.
21. Schneier, B., Mudge, P., “Cryptanalysis of Microsoft’s Point-to-Point Tunneling Protocol (PPTP)”, Counterpane Systems, Minneapolis MN, 1999.
22. Robin, J. *Analyzing the Intel Pentium’s Capability to Support a Secure Virtual Machine Monitor*, Naval Postgraduate School, Monterey, CA, September 1999.
23. *Novell Trusted Workstation Partnership*, First Edition, Novell Network Security Development, October 1998.
24. *Media Encryption Management System: System Architecture Document*, Revision 1.5, Spyrus, San Jose, CA, 5 August 1996.
25. *Media Encryption Management System: System Architecture Document Addendum* Revision 1.0, Spyrus, San Jose, CA, 8 November 1995.
26. Wang, R. *Flash in the Pan?*, <http://www.virusbtn.com/VirusInformation/cih.html>, 1999.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center 8725 John J. Kingman Rd., Ste 0944 Ft. Belvoir, VA 22060-6218	2
2.	Dudley Knox Library Naval Postgraduate School 411 Dyer Rd. Monterey, CA 93943-5101	2
3.	Chairman, Code CS Computer Science Department Naval Postgraduate School Monterey, CA 93943-5000	1
4.	Dr. Cynthia E. Irvine Computer Science Department Code CS/Ic Naval Postgraduate School Monterey, CA 93943-5000	3
5.	Mr. James P. Anderson James P. Anderson Company Box 42 Fort Washington, PA 19034	1
6.	Mr. Paul Pittelli National Security Agency Research and Development Building R2, Technical Director 9800 Savage Road Fort Meade, MD 20755-6000	1
7.	CAPT Dan Galik Space and Naval Warfare Systems Command PMW 161 Building OT-1, Room 1024 4301 Pacific Highway San Diego, CA 92110-3127	1

8. Commander, Naval Security Group Command..... 1
 Naval Security Group Headquarters
 9800 Savage Road
 Suite 6585
 Fort Meade, MD 20755-6585

9. Mr. George Bieber..... 1
 Defense Information Systems Agency
 Center for Information Systems Security
 5113 Leesburg Pike, Suite 400
 Falls Church, VA 22041-3230

10. Ms. Louise Davidson 1
 N643
 Presidential Tower 1
 2511 South Jefferson Davis Highway
 Arlington, VA 22202

11. Mr. William Dawson..... 1
 Community CIO Office
 Washington DC 20505

12. Ms. Deborah M. Cooper..... 1
 Deborah M. Cooper Company
 P. O. Box 17753
 Arlington, VA 22216

13. Mr. Robert Wherley..... 1
 XTS Product Technical Manager
 WANG Federal Inc.
 7900 Westpark Drive
 McLean, VA 22102-4299

14. Mr. Paul Barbieri..... 1
 WANG Federal Inc.
 7900 Westpark Drive
 McLean, VA 22102-4299

- 15. COL Timothy A. Fong..... 1
COMMANDER/JED/IAESO
Columbia Pike Offices
5600 Columbia Pike
Falls Church, VA 22041-2717

- 16. LCDR James P. Downey 1
DISA D6/IAESO/MSL Engineering
5600 Columbia Pike
Falls Church, VA 22041-2717

- 17. LT Steven R. Balmer 1
393 A Ricketts Rd.
Monterey, CA 93940