The Center for Information Systems
Security Studies and Research

Technical Report NPS-CS-05-001

Trusted Computing Exemplar

2004 Developments

Cynthia E Irvine, Timothy E. Levin, Thuy D. Nguyen
October 2004

ACKNOWLEDGMENTS

This work was sponsored in part by the Office of Naval Research grants N001403AF00002, N0001403WX21224, and N001404WR20357.
This work was funded in part by the National Reconnaissance Office under funding document E338270.
Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Office of Naval Research or the National Reconnaissance Office.

1 Introduction

The Center for Information Systems Security Studies and Research at the Naval Postgraduate School (CISR) has established and is vigorously pursuing a multifaceted research and development project to provide an openly distributed worked example of how high assurance trusted computing components can be built. The *Trusted Computing Exemplar* project (TCX) encompasses four related activities: creation of the High Assurance Rapid Development Environment (HARDE) prototype; development of a reference-implementation trusted computing component (the TCX Separation Kernel); evaluation of the component for high assurance; and open dissemination of results related to the first three activities.

This document presents a brief overview of the TCX Separation Kernel design (see Section 2), and provides a summary of TCX developments that have occurred in financial year 2004 – October 2003 through September 2004. Significant progress on the development environment is described in Sections 3 and 4. System development and evaluation progress are described in section 5. Our high-level design requirements have been defined synergistically with a Common Criteria protection profile for separation kernels. Work on open dissemination of TCX project results is included in "Documentation Integration Environment," in Section 2. Portions of that environment will eventually be integrated into the dissemination system.

2 Background

2.1 TCX Project Goals and History

Over the past decade, neither the private sector nor the US Government have been significantly involved in high assurance Trusted Computing acquisitions and research. During this time, the focus on commercial-off-the-shelf procurements by these sectors has helped to fuel explosive advances in commercial technology, but it also contributed to the lack of progress in the ability of commercial systems to appropriately protect themselves and the data with which they are entrusted. No new high assurance (viz. TCSEC Class A1 or Common Criteria EAL6/7) systems have been fielded. While industry has been driven to supply the latest technology at the fastest pace, it has not been motivated, either internally or externally via customer demand, to produce highly trustworthy computing systems. As a result, the National Information Infrastructure is weak; there are no contemporary high security, high assurance, off-the-shelf products available that can be used to strengthen it; and the National capability to design and construct such trusted computer systems and networks has atrophied. To help address these shortcomings, we have established the Trusted Computing Exemplar (TCX) project.

The concept of TCX was developed in 2002 and 2003. Project work commenced in 2004, with a combination of funding from several national sources (see Appendix B). Today, we have more than 14 research, engineering and support personnel, as well as many graduate students, working on the TCX project.

2.2 TCX Design Overview

Our approach is to develop a high assurance separation kernel, along with a trusted application built to be hierarchically layered [dijkstra68] on the Kernel, as a reference implementation exemplar system for trusted computing. The high-level requirements of the TCX Separation Kernel (TCX-SK) are for a small, portable component that will take advantage of modern hardware support, where applicable, and that will provide users with correct security operation and an *a priori* assurance against system subversion. To demonstrate the Kernel's utility, a high assurance network authenticator will be developed as the application portion of the Trusted Computing Exemplar system.

2.2.1 TCX Kernel

The primary security function of TCX-SK will be to enforce process and data-domain separation, while providing primitive operating system services sufficient to support simple applications. The embedded focus of the kernel drives several high-level design characteristics. The kernel will be small but complete with respect to policy enforcement. It will have a static runtime resource configuration and its security policy regarding access to resources will be based on static process/resource access bindings, which are subject to offline configuration. We anticipate that the kernel will support a small number of processes, data objects, and I/O devices. A "RAM disk" can be constructed from storage resources, but there is no plan to support a hard drive. Below, we provide a few more details about our view of the kernel.

Application processes will be scheduled in a round robin fashion with each process being given a predetermined amount of time, set by the configuration. The process/resource access binding mechanism within the kernel will allow the assignment of specific modes of access, such as modify and observe, by which a process may access a particular resource. A variety of policies can be represented by this method, including one indicative of a lattice of security domains [denning76a].

The kernel will provide mechanisms to handle asynchronous interrupts. Any I/O will be handled by the kernel and presented to processes via memory segments.

Processes may communicate through shared memory segments that at least one can write and the other can read, or through simple process synchronization primitives, which can be implemented to be demonstrably free of covert channels [reed79, levin90].

The static nature of resource allotment allows the TCX-SK to both minimize internal runtime activity that could form the basis of covert channels between processes [kemmerer82], and to provide more predictable processing behavior, than would otherwise be possible.

2.2.2 Demonstration Application: the Trusted Path Extension

The demonstration system - a demonstration application running on the TCX-SK, including a small form-factor hardware base - is a high assurance network authentication device, for communicating security critical information between a user and a remote secure server, such as in the MYSEA distributed security architecture (see Figure 1 [irvine04b]). This *trusted path extension* (TPE) device interfaces between a specially configured COTS workstation and the network. The TPE application provides authentication and session negotiation services with which users can establish trusted

sessions with the server, as well as cryptographically protected communications between the TPE and the secure server. Once logged on through TPE, user sessions at the workstation may interact with the secure server, as limited by the negotiated security parameters (e.g., military session level, or enterprise group membership). For this architecture, the workstation does not need to be trusted to support security critical services.

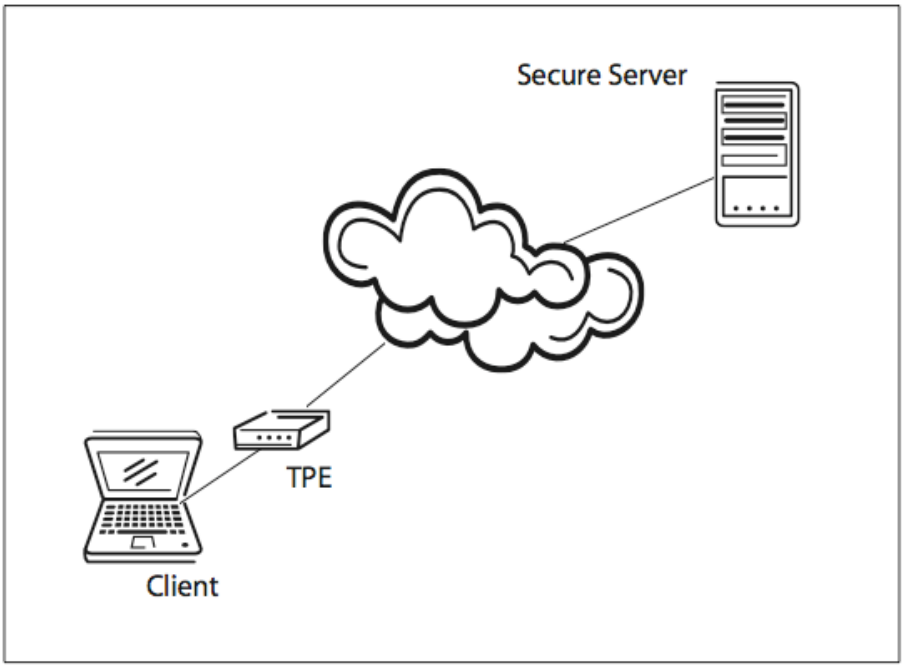

Figure 1. MYSEA System Architecture: Client-TPE-Server.

Hosting of the TPE on a separate, small form-factor device provides several benefits. First, the device processor and memory are physically separate from the workstation, so that there is no question of interference from user processes. Second, trusted path functions need to have absolute control of the screen and keyboard to ensure that the user is communicating directly with the trusted computing base. Since the TPE device has its own screen and keyboard, we can ensure that trusted path I/O functions are completely controlled without having to analyze the potentially complex mechanisms of the workstation screen and keyboard. Third, the *secure attention key* design, through which the user signals that s/he desires to communicate with the TCB, is straightforward relative to attempting to utilize keystrokes from a standard workstation keyboard.

3 Product Life Cycle

The product life cycle determines many aspects of what, when, how, and by whom the project is to proceed. During 2004, a set of TCX life cycle-related documents was completed, as follows.

3.1.1 Life Cycle Plan

The overall purpose and guiding principle for this document is to provide a methodology that will result in the creation of a product that will have a high level of assurance that its security policies are sound and correctly implemented. A secondary purpose for this

document is to ensure compliance with the Common Criteria (CC) requirements for Evaluation Assurance Level (EAL) 7. [cc04] This plan is the overarching high-level document that guides a product throughout its life cycle, from its initial conception to its eventual retirement, by defining policy, process, and high-level procedures.

3.1.2 Configuration Management Plan

This document describes the procedures and policy for Configuration Management (CM) of TCX. This document and the measures it describes are also intended to satisfy the Common Criteria (CC) Configuration Management requirements for EAL6 and above. The objectives of this CM plan include: ensuring the integrity of the configuration items, tracking changes to the configuration items, and ensuring only authorized changes are made to the configurations items. The items to be managed under this plan include any documents, source code, specifications, and other items written, used or developed, including bug reports, security flaws, and development tools, as part of the product development process.

3.1.3 Configuration Management Procedures

The purpose of this document is to outline the procedures for the Configuration Management (CM) process. These procedures are meant to provide lower-level details necessary to implement the process laid out in the Configuration Management Plan, and to ensure consistency in the exercise of the process. Additional procedures are provided to interface with CM-specific applications. Drafts of the CM system design and assembly document, and the CM system user's guide were completed and are already in use to train CM staff.

3.1.4 Physical and Personnel Security Plans

Because of the nature of the project, integrity is the primary policy concern of these plans, though confidentiality is not disregarded (e.g., with respect to unintended dissemination of incomplete articles).

- The purpose of the Physical Security Plan is to define policies necessary to ensure the physical protection of the TCX during its entire life cycle.
- The purpose of the Personnel Security Plan is to define personnel policies necessary to protect the confidentiality and integrity of the TCX data during its entire life cycle.

4 High Assurance Rapid Development Environment

The TCX High Assurance Rapid Development Environment (HARDE) consists of a documentation integration environment with which to define and describe TCX project documents, development tools and procedures with which to build TCX software, and verification tools with which to determine with high assurance whether the system that is built is as was defined. Significant progress was made in all of these areas in 2004.

4.1 Software Development Tools, Procedures and Standards

We have completed the requirements analysis, design, and construction of a prototype for the software development environment. Setup procedures for the software development environment are being defined. We have also completed the software development

standards document, which includes (1) processes for designing, approving and developing Configuration Items in conformance with the CM Plan, as well as (2) the TCX coding standards.

The prototype development server was assembled to run Windows Server 2003 Standard Edition software and the OpenWatcom compiler. Support for distributed development was also configured, to be accessible through both Linux and Windows clients.

4.2 Documentation Integration Environment

We have established initial high-level requirements for the Documentation Integration Environment (DIE). These requirements include the following:

- The DIE must provide or support interfaces, schemas, and templates suitable for all document types and formats required by the Documentation Standards.
- The DIE must support a hierarchy of documentation resources, as well as the ability to map specific requirements and product features at one level of the hierarchy to requirements and features at another level.
- The DIE must be able to validate the proper conformance of resources to project schemas and other standards.
- The DIE must provide a mechanism for extracting project resources by name, resource ID, and version from the CM system to individual workspaces, as well as an interface for checking in modified resources to the CM system.
- The DIE must provide on-request invocation of testing, compiling and verification tools, while maintaining the integrity of the original resource.
- The DIE must provide both current and published views of the documents in the project. These views must be identical to the designated current and published versions maintained by the CM system.

The specifications and implementation of the initial version of the DIE have been completed, and the DIE is in use by the project team for the creation and requirements mapping of project documents.

4.3 Verification Tools

In 2004, the TCX verification tool requirements were defined, the verification server and several candidate tool sets were assembled, and analysis proceeded toward selection of the verification tool set for the project.

The purpose of the TCX verification tool requirements document is to describe the requirements for the software tools that will be used for the verification of the TCX Kernel. Verification includes creation of the following “system representations:” system security policy, system functional requirements, formal security policy model, formal high-level design specification for each subsystem, semi-formal low-level design specification for each subsystem.

Verification also involves various forms of analysis regarding the system representation documents, such as covert channel analysis, formal proof of the model and formal, semi-formal or informal proof of correspondence between different representations (depending on the level of formality of the particular representations).

Tools and requirements for automated support of the verification analyses have been defined, including:

- Specification language syntax checker
- Theorem prover
- Requirements mapping system (part of DIE, above)
- Non-determinism checker
- Information flow analyzer
- Shared resource matrix generator

We have procured and configured a server to act as the verification server, and have installed several verification tool suites onto it, including FDM, PVS, and ACL2. Investigation and experimentation is ongoing to determine suitability of each tool suite for the verification of high assurance systems models and specifications.

5 System Design and Development

The high-level policy and requirements definition for the TCX-SK has been completed. These requirements have been developed in conjunction with the generic requirements definition (viz., Protection Profile) of a class of high assurance separation kernels [nsa04]. Since the TCX project will be able to utilize the Protection Profile during the TCX-SK evaluation process, our participation in the development of that profile provides progress for both our system design activity and the product evaluation activity.

For Common Criteria evaluation, a security target document is required that provides further detail about the implementation than does the Protection Profile. We have produced a first draft of the TCX-SK Security Target. The security target will help guide the other system design documents (see "system representation documents," under Verification Tools, above).

Finally, in conformance with our configuration management plan (see "Product Life Cycle," above), we have completed initial versions of the TCX Project Configuration Item List, and the TCX-SK Configuration Item List.

6 Future Plans

In the next year we expect to make significant progress in several TCX project activities. In the TCX High Assurance Rapid Development Environment (HARDE) area, we will transition the software development environment into use for TCX development. The DIE will be enhanced with further functionality, and be integrated with the CM system. Also, final selection and configuration of the verification tools will occur.

In the System design and development area, we expect to formulate the system architecture and system functional requirements next year, as well as begin work on the formal security policy model and formal high-level design specifications. We will also develop the high-level policy and requirements definition for the TPE.

Finally, we will design and prototype the dissemination system, to include access to the documentation structure of the DIE.

REFERENCES

cc04 Common Criteria for Information Technology Security Evaluation, Version 2.2, CCIMB-2004-01-00[1, 2, 3], January 2004

denning76a Dorothy E. Denning, A Lattice Model of Secure Information Flow, Communications of the A.C.M., Volume 19, Number 5, Pages 236-243, 1976

dijkstra68 Dijkstra, Edsger W., The Structure of the "THE"-Multiprograming System, Communications of the A.C.M., Volume 11 Number 5, Pages 341-346, 1968

Irvine04a Irvine, Cynthia E., Levin, Timothy E., Nguyen, Thuy D., and Dinolt, George W., The Trusted Computing Exemplar Project, Proceedings of the 2004 IEEE Systems, Man and Cybernetics Information Assurance Workshop, West Point, NY, June 2004, pp. 109-115.

irvine04b Cynthia E. Irvine, Timothy E. Levin, Thuy D. Nguyen, David Shifflett, Jean Khosalim, Paul C. Clark, Albert Wong, Francis Afinidad, David Bibighaus and Joseph Sears, Overview of a High Assurance Architecture for Distributed Multilevel Security, Proceedings of the 2004 IEEE Systems, Man and Cybernetics Information Assurance Workshop, West Point, NY, June 2004.

kemmerer82 Author Richard A. Kemmerer, A Practical Approach to Identifying Storage and Timing Channels, Proceedings of the 1982 IEEE Symposium on Security and Privacy, Pages 66--73, Oakland, CA, April 1982

levin90 Timothy E. Levin and Albert Tao and Steven J. Padilla, Covert Storage Channel Analysis: A Worked Example, Proc. National Computer Security Conference, Pages 10-19, Washington, DC, October 1990

nsa04 U.S. Government Protection Profile for Separation Kernels in Environments Requiring High Robustness, National Security Agency, 1 July 2004, http://niap.nist.gov/pp/draft_pps/pp_draft_skpp_hr_v0.621.html

reed79 D.P. Reed and R.K. Kanodia, Synchronization with Eventcounts and Sequencers, Communications of the A.C.M., Volume 22 Number 2, Pages 115-123, 1979