NPS-CS-07-001

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

SecureCore Software Architecture:
Trusted Path Application (TPA) Requirements

by

Paul C. Clark
Cynthia E. Irvine
Timothy E. Levin
Thuy D. Nguyen
Timothy M. Vidas

December 2007

Approved for public release; distribution is unlimited

This page intentionally left blank

NAVAL POSTGRADUATE SCHOOL
Monterey, California 93943-5000

Vice Admiral Daniel T. Oliver (Retired)
President

Leonard Ferrari
Provost

This material is based upon work supported by the National Science Foundation (NSF). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of that agency.

This report was prepared by:

Paul C. Clark
Research Associate

Cynthia E. Irvine
Professor

Timothy E. Levin
Research Associate Professor

Thuy D. Nguyen
Research Associate

Timothy M. Vidas
Research Associate

Reviewed by:

Peter J. Denning, Chair
Department of Computer Science

Released by:

Dan C. Boger
Interim Associate Provost and
Dean of Research

This page intentionally left blank

REPORT DOCUMENTATION PAGE

Form approved

OMB No 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	10 December 2007	Research; September 2006 - December 2007

4. TITLE AND SUBTITLE

SecureCore Software Architecture: Trusted Path Application (TPA) Requirements

5. FUNDING

CNS-0430566

6. AUTHOR(S)

Paul Clark, Timothy E. Levin, Cynthia E. Irvine, and Thuy D. Nguyen, Timothy M. Vidas

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Postgraduate School
Center for Information Systems Security Studies and Research (CISR)
1411 Cunningham Road, Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER

NPS-CS-07-001

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Science Foundation (NSF)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

This material is based upon work supported by the National Science Foundation (NSF). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words.)

A mobile computing device has more inherent risk than desktops or most other stationary computing devices. Such mobile devices are typically carried outside of a controlled physical environment, and they must communicate over an insecure medium. The risk is even greater if the data being stored, processed and transmitted by the mobile device is classified. The purpose of the SecureCore research project is to investigate fundamental architectural features required for the trusted operation of mobile computing devices so the security is built-in, transparent and flexible. A high-level architecture is described to provide such features. In addition, a usage scenario is described for a potential use of the architecture, with emphasis on the trusted path, a non-spoofable user interface to the trusted components of the system. Detailed requirements for the trusted path are provided.

14. SUBJECT TERMS

Access Control, Trusted Path, High Assurance, Security Kernel

15. NUMBER OF PAGES

24

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unclassified

This page left intentionally blank

NPS-CS-07-001

Trustworthy Commodity Computation and Communication

SecureCore Technical Report

SecureCore Software Architecture:
Trusted Path Application (TPA) Requirements

Paul C. Clark, Cynthia E. Irvine, Timothy E. Levin, Thuy D. Nguyen, Timothy M. Vidas

December 10, 2007

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. CNS-0430566 and CNS-0430598 with support from DARPA ATO. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or of DARPA ATO.

Author Affiliation:

Center for Information Systems Security Studies and Research
Computer Science Department
Naval Postgraduate School
Monterey, California 93943

Table of Contents

List of Figures

List of Tables

1 Introduction

SecureCore is a research project funded by the National Science Foundation (NSF) to investigate the fundamental architectural features required for trustworthy operation of mobile computing devices such as smart cards, embedded controllers and hand-held computers. The goal is to provide secure processing and communication features for resource-constrained platforms, without compromise of performance, size, cost or energy consumption. In this environment, the security must also be built-in, transparent and flexible.

The significance of this research lies in the exploration of new approaches for threat and requirements analysis, a fresh look at integrated support for security, performance, functionality and usability in mobile platforms, and the potential for innovative advancements in processor instruction set architecture, operating system kernel design, and secure network protocols. The SecureCore architecture is described in a set of documents; this document is a member of that set.

As a guide for the SecureCore research, a concept-of-operations (CONOPS) has been developed that focuses on the use of handheld computing devices by emergency-response personnel. It is generally understood that many emergencies could be handled more effectively if vital information (e.g., schematics of a large burning building) could be communicated in a timely and secure manner to the responders [1]. Emergencies in the context of the SecureCore CONOPS are assumed to be major disasters that involve vast resources from various government and non-government organizations, with one organization acting as a coordinating central authority (e.g., the Department of Homeland Security (DHS)). The central authority would establish memoranda of understanding (MOUs) with other agencies in advance of an emergency. The MOU would be a two-way agreement: on one hand it would establish the ability for the central authority to request information from its partner agencies during an emergency, and on the other hand it would establish minimum security requirements imposed by the third party for the handling of the data when it is provided to the central authority.

Functional requirements for a SecureCore handheld include the ability to support both a “normal context” and a “trusted context”. A rough definition of a normal context is the ability to interact with the handheld in a way that is similar to any commercial handheld or Personal Digital Assistant (PDA). A rough definition of a trusted context is the ability to use the same handheld in a high assurance fashion that can be relied upon to handle sensitive data and user interactions properly. This is shown conceptually in Figure 1.

Figure 1. Normal and Trusted Contexts

To provide the normal and trusted contexts a Least Privilege Separation Kernel (LPSK) will support a partitioning of handheld resources (e.g., disk space and memory), such that guest operating systems (OSs) can be multitasked without allowing covert information flows between them. To keep the kernel small and simple, the SecureCore Security Services (SCSS) layer provides other handheld-wide security services. Both the LPSK and the SCSS are trustworthy components, and are referred to collectively as the Trusted Management Layer (TML). These can both be seen in Figure 2, where white areas are shown as trustworthy components, and where the vertical separations above the kernel represent partitioned resources.

Figure 2 also shows a trusted component supported by the SCSS known as the SecureCore operating system (SCOS), with the trusted path application (TPA) as one of its applications. The SCOS provides minimal high-level operating system-like services to the applications that run on top of it. The TPA is an application that provides a trusted user interface between the user and the other trusted components of the handheld.

It is expected that the SecureCore handheld will store data with varying levels of classification (e.g., UNCLASS and SECRET), and support users with varying levels of clearance, which would classify it as a multilevel-secure (MLS) handheld. Therefore, the trustworthiness of the trusted components will need to be very high. This high level of trust will be accomplished by following recognized security design principles [2] and recognized security evaluation standards, such as the Common Criteria [3].

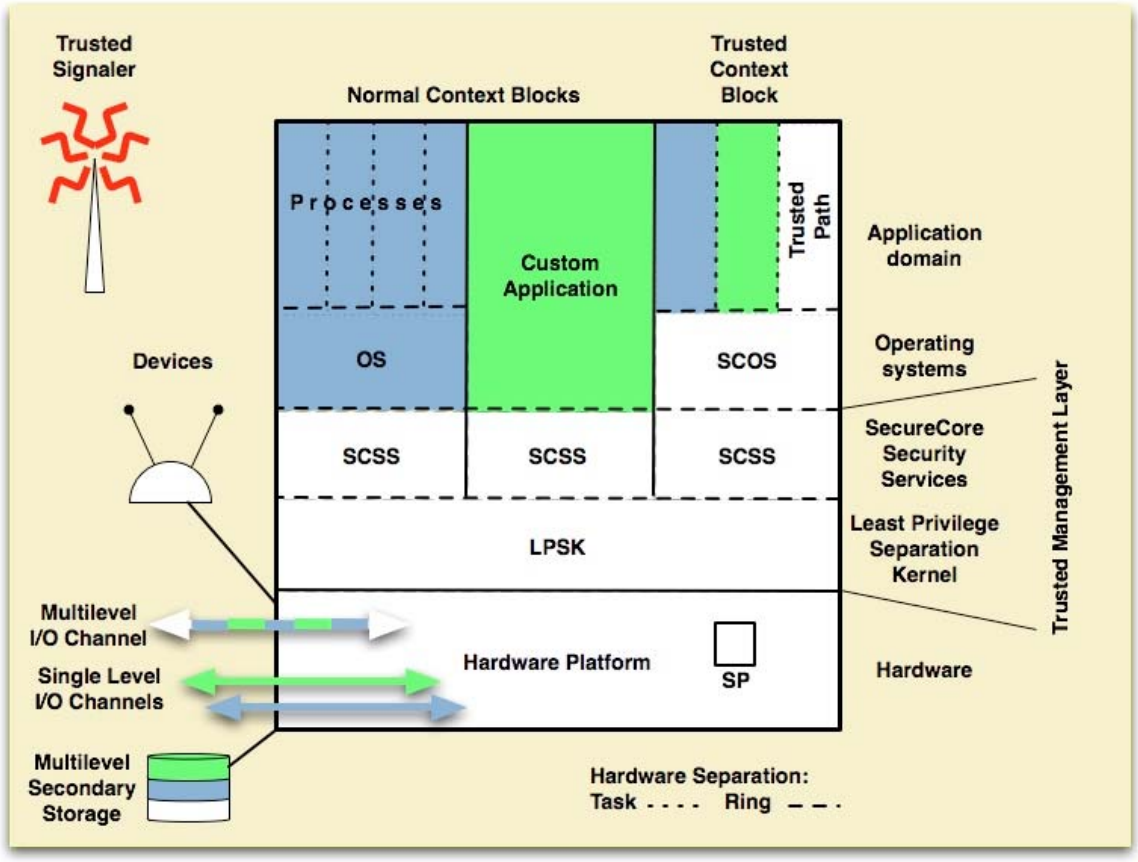

Figure 2. Software Architecture

Table 1 lists the assignment of functions and policies to the various components of the Secure handheld.

Layer	Functions and Policies
TPA	Trusted Path interface to security-critical services
SCOS	Application Management Identification and Authentication Operating System Services
SCSS	MLS Support and Interpretation Resource Virtualization Object Management Focus Management Trusted Channel Management Inter-Partition Routing
LPSK	Partitioning of Resources Resource Management MAC Enforcement Partition Scheduling Cross-Partition and Inter-Process Communication

Table 1. Function and Policy Assignment

The SecureCore handheld will support both confidentiality and integrity policies. Subjects and objects will be associated with security labels that identify their combined confidentiality and integrity session levels or classifications. When referring to such labels this document will use the A:B syntax, where A is the confidentiality component of the label and B is the integrity component of the label. An example of such a label is UNCLASS:LOW. Each component will include a hierarchical level and optional non-hierarchical compartments.

As mentioned above, this project is meant to produce a proof-of-concept for a computing platform that is suitable for use "in the field", and is envisioned to be a small form-factor device, and is hereafter referred to as the *SecureCore handheld*. Besides the features and security goals of the SecureCore handheld that have already been described, the following non-security factors were considered to determine the scope and functionality of the initial device:

1. Cost of materials
2. Time to design and implement
3. Military and non-Department of Defense (DoD) government and first responder needs
4. Usability
5. Efficient use of resources, such as power usage

2 Glossary

This section provides definitions of generally accepted security terms that are independent of this project, and which are referenced in this document. They are listed in order of dependency.

Trusted Computing Base (TCB)

> "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." [4]

Trusted Path

> "A mechanism by which a person at a [computer] can communicate directly with the Trusted Computing Base. This mechanism can only be activated by the person or the Trusted Computing Base and cannot be imitated by untrusted software". [4]

Secure Attention Key (SAK)

> A secure attention key is a special hardware mechanism for invoking the Trusted Path. For example, it can be a reserved key sequence on a keyboard to be invoked by a user to signal the Trusted Computing Base that communication between them needs to occur.

Security Perimeter

> "The boundary where security controls are in effect to protect assets." [4]

Subject

> "An active entity, generally in the form of a person, process, or device that causes information to flow among objects or changes the system state." [4]

3 Definitions

To minimize confusion while reading this report, the following definitions are provided. The definitions are loosely ordered by dependency, such that the later definitions are dependent on the earlier definitions. The word "platform" is used to refer to a SecureCore handeld instead of "device" to avoid confusion when referring to components of the handheld, such as the network card or keyboard.

3.1 Power States

Platform Initialization

Platform initialization relates to a "hard" boot of a SecureCore handheld. Initialization on a desktop computer happens on a fairly regular basis (e.g., daily), whereas initialization on a Personal Digital Assistant (PDA) rarely happens (e.g., during the first power-up of the device, and when the system hangs and is reset). The initialization of a SecureCore handheld includes the initialization of all trusted components of the system (i.e, the LPSK, followed by the SCSS, followed by all SCOS instantiations, followed by the TPA) and may include the *initiation* of the boot sequence for guest operating systems that may exist in active partitions outside of the security perimeter.

Platform Shutdown

A SecureCore handheld is considered shut down when all subjects have been halted and the system powered off, such that additional processing requires a platform initialization. Platform shutdown consumes the least power of any other state.

Platform Standby

Standby is a state where a SecureCore handheld is put into a low-power-usage mode (e.g., hard disks stop spinning, I/O is terminated), but the system is still available for nearly instantaneous access. Desktops and laptops may use this mode, but PDAs depend on it.

Platform Hibernation

Hibernation is when the state of the SecureCore handheld is saved to hard disk prior to a platform shutdown. When the platform is turned on, the saved state is restored from hard disk, rather than ending at the normal initial state after platform initialization.

Figure 3 shows a power state diagram and how the above definitions relate to each other.

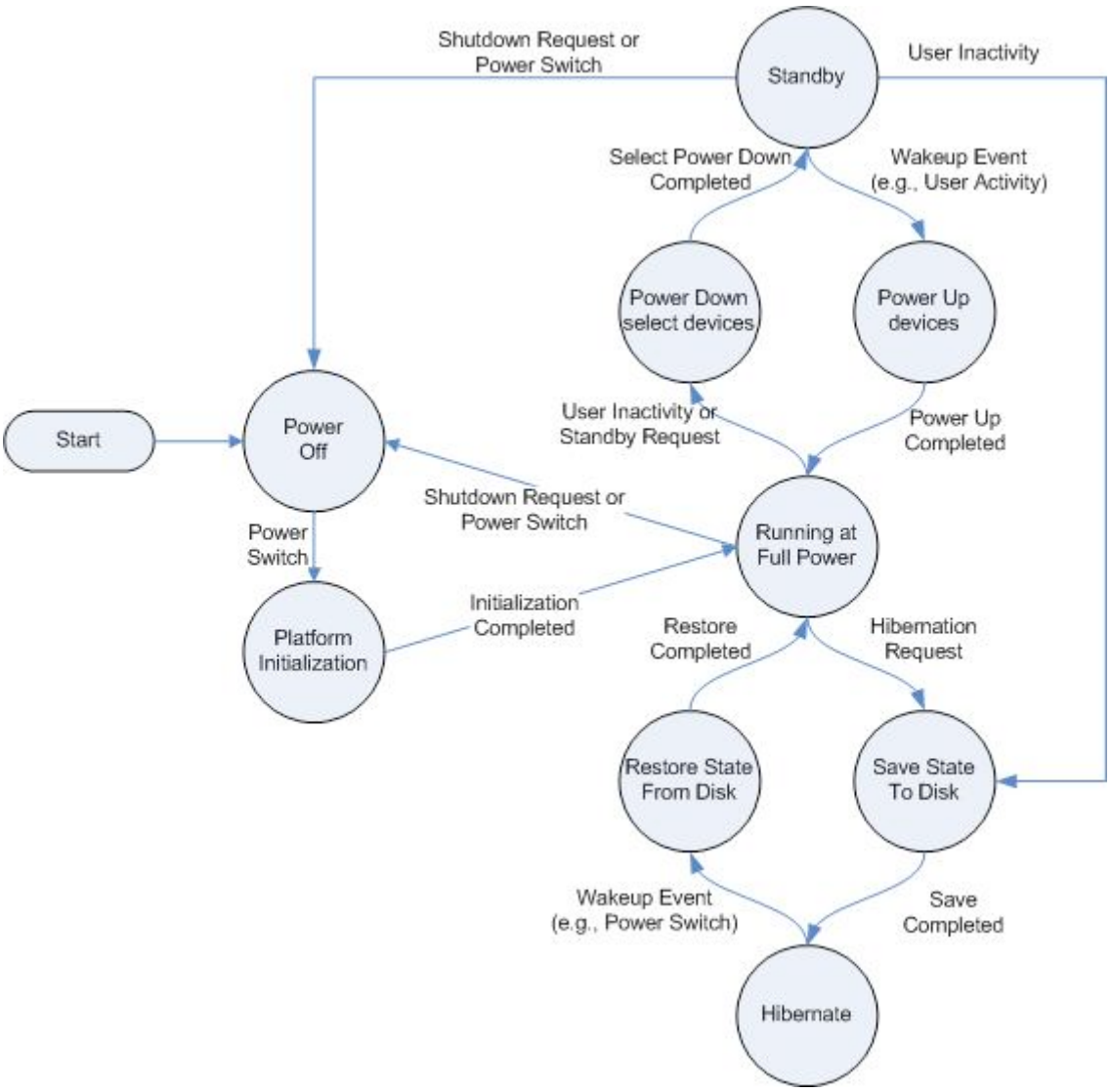

Figure 3. Power State Diagram

3.2 Miscellaneous Definitions

Trusted Management Layer (TML)

The Trusted Management Layer is a term that refers to the LPSK and the SCSS collectively.

Trusted Communication Channel

A trusted communication channel is a TCB-to-TCB communication path that provides confidentiality and integrity for the traffic carried between them, and where the subjects involved in the communication have authenticated each other.

Remote Trusted Path

A remote trusted path is a mechanism to establish communication with the TCB of a remote system. The remote trusted path is invoked by a person interacting with a local TCB. The local TCB, in turn, establishes a trusted communication

channel with the remote TCB to support the remote trusted path. As with a local trusted path, a remote trusted path "cannot be imitated by untrusted software" [4].

Remote Secure Attention Key (Remote SAK)

A remote SAK is the first step in a network-based protocol that is used to establish a remote trusted path.

Emergency Signal

An emergency signal is a network-based communication from a recognized central authority (over a trusted communication channel) that an emergency situation has been declared.

Emergency Security Level

The emergency security level is the security level associated with a declared emergency.

3.3 Partition-Related Definitions

Partition With(out) Focus

A partition has *focus* if it has access to the keyboard and screen, such that the user may interact with it. All other partitions are referred to as *partitions without focus*. Only one partition can have focus at a given time.

Current Executing Partition

The current executing partition is the partition assigned the current CPU time slice. The current executing partition can be a partition without focus.

Active and Passive Partitions

An *active partition* is a partition that the LPSK schedules for the potential execution of subjects that may have been instantiated. If a partition is not scheduled by the LPSK, i.e., it is intended to only hold data, then the partition is known as a *passive partition*.

Normal Partition

A *normal* partition is an active partition that contains low-assurance subjects outside of the TML, e.g., a commodity operating system (OS) or specialized application.

Trusted Partition

A *trusted* partition is an active partition that contains high assurance subjects outside of the TML.

Emergency Partition

The emergency partition is a partition that is configured at a security level that was pre-determined by a central authority (and third party data providers) for the proper handling of sensitive data during a declared emergency situation. Such partitions are available for focus only during such emergencies, as described in Section 5. A user is not limited to the emergency partition(s) during an emergency; rather, an emergency situation makes these additional partition(s) available for focus change.

TPA Partition

The TPA partition is the trusted partition that contains the TPA. There can only be one TPA partition per SecureCore handheld.

Default Partition

The default partition is the partition that is configured to be the partition with focus after platform initialization has been completed. If the default partition is not explicitly defined in the installed configuration data (e.g., the SCSS configuration data), then the default partition is the TPA partition. Partitions configured to be the default partition must be either the TPA partition, or a partition with a security level of UNCLASS:LOW.

Guest Operating System

A guest operating system (OS) is any OS that is running on top of the TML, whether in a normal partition or a trusted partition.

Partition Halt

A partition halt refers to the stopping of all non-TML subjects within a given partition. From the point of view of the non-TML subjects it is equivalent to a sudden loss of power.

Partition Boot

A *partition boot* refers to the starting of the initialization of a non-TML subject for a given partition. From the point of view of the non-TML subject (e.g., a guest OS) it is equivalent to turning on the power to the "computer" it is installed on.

3.4 Authentication-Related Definitions

TPA Login

TPA login refers to logging into the TPA via a trusted path. A successful TPA login provides additional menu options from the TPA that require Identification and Authentication (I&A) before being used, such as switching focus to partitions that are classified above UNCLASS:LOW.

TPA Logout

TPA logout occurs when the *logout* option is selected from a TPA menu. TPA logout limits the TPA menu options to those that do not require I&A.

Platform Lock

Platform lock is a state that excludes all user I/O until the SAK is invoked and the user enters the correct password via the TPA. A platform lock can only occur if a user has performed a TPA Login. The state is entered either explicitly via the TPA menu or by a configured length of inactivity. If the state was entered via a menu choice, entering the correct password unlocks the platform and displays the proper TPA menu. If the state was entered via inactivity, then entering the password returns the focus to the partition that was in focus prior to the platform lock.

3.5 Context-Related Definitions

Normal Context

Normal context refers to a user interaction with a SecureCore handheld normal partition.

Trusted Context

Trusted context refers to a user interaction with a SecureCore handheld trusted partition.

Transient Trust

Transient trust refers to **temporarily** trusting a user with data objects that are urgently needed. For example, under non-emergency circumstances the user would not have privilege or clearance to view the objects, but an emergency situation may allow the user to view them only during the declared emergency. In this example, a system supporting transient trust would need to revoke access to the objects shared under such conditions when the emergency situation is over.

4 Phased Implementation

There are currently four phases planned for the design and implementation of a SecureCore handheld. These phases are described below.

4.1 Phase 0

This is a rapid prototype without a lot of engineering put into interfaces and layering, and with a lot of the planned functionality missing. It will be built on existing Intel x86 hardware without the use of hardware-assisted virtualization features that may be available. The entire TML will be in PL0, the SCOS will be in PL1, and a demo application will be in PL1 or PL3, as shown in Figure 4.

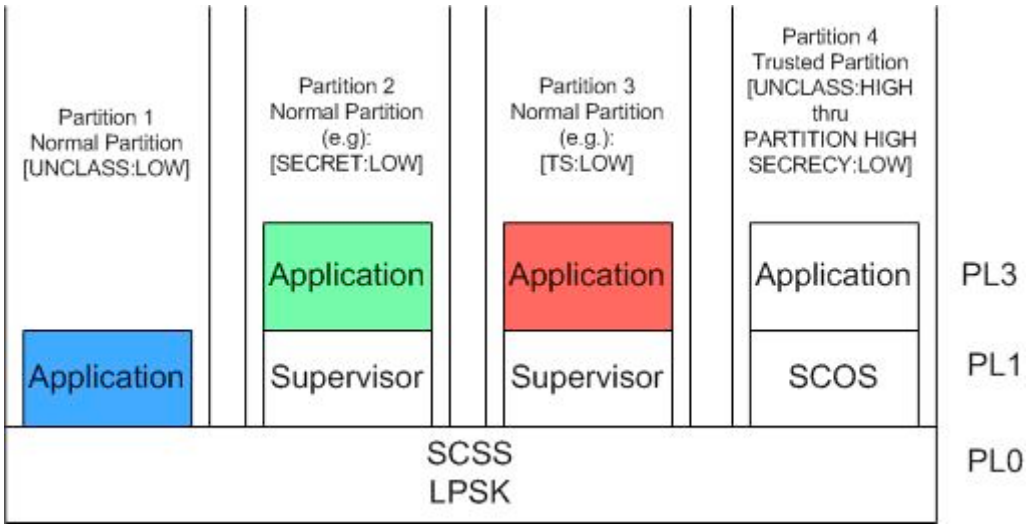

Figure 4. Phase 0 Architecture

No guest operating systems will be supported. However, a *supervisor* will be developed to provide services to applications in low-integrity partitions. The supervisor will provide a subset of operating system services, such as a library for easily providing user I/O with the screen and keyboard. Applications will run in their own partition using either an SCOS interface, the supervisor, or directly using the SCSS interface.

The only devices virtualized and shared between partitions are the keyboard and screen. No networking will be supported. Only a crude TPA will be provided, if any.

4.2 Phase I

The requirements listed in this document relate to Phase I.

Drafts of the high-level specifications for all the planned SecureCore functionality will be completed, influenced by the experience of working on Phase 0. The specifications will assume the availability of only four x86 privilege levels, and will be allocated as follows: LPSK in PL0, SCSS in PL1, SCOS in PL2, and the TPA in PL3, as shown in Figure 5.

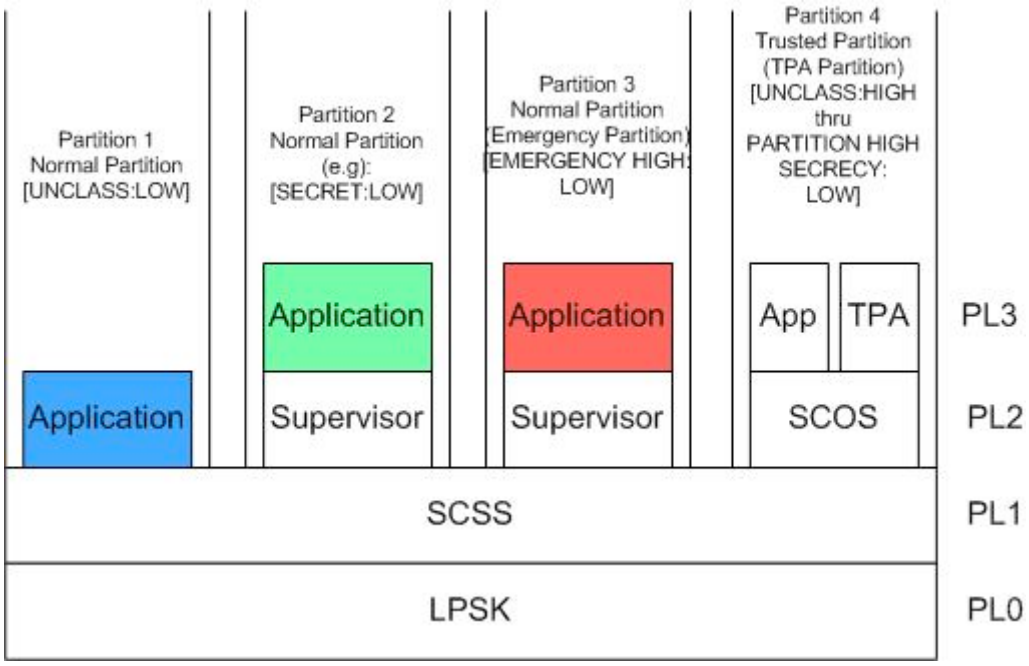

Figure 5. Phase I Architecture

The SCSS will provide a virtualized network card that can be accessed by all partitions. The emergency partition with its transient trust capabilities will be supported by an emergency application on top of a supervisor (or an instantiation of the SCOS). The TPA will be able to activate applications within its partition via the SCOS interface.

The prototype will be modified to provide the specified functionality, but it may not comply with the specification.

4.3 Phase II

Phase II will produce a rapid prototype of the SecureCore architecture using hardware-assisted virtualization features of an x86-based CPU. The PL architecture of the demonstration software will change so that the LPSK is in PL-2, the SCSS is in PL-1, the SCOS is in PL0, and the TPA is in PL1, as shown in Figure 6.

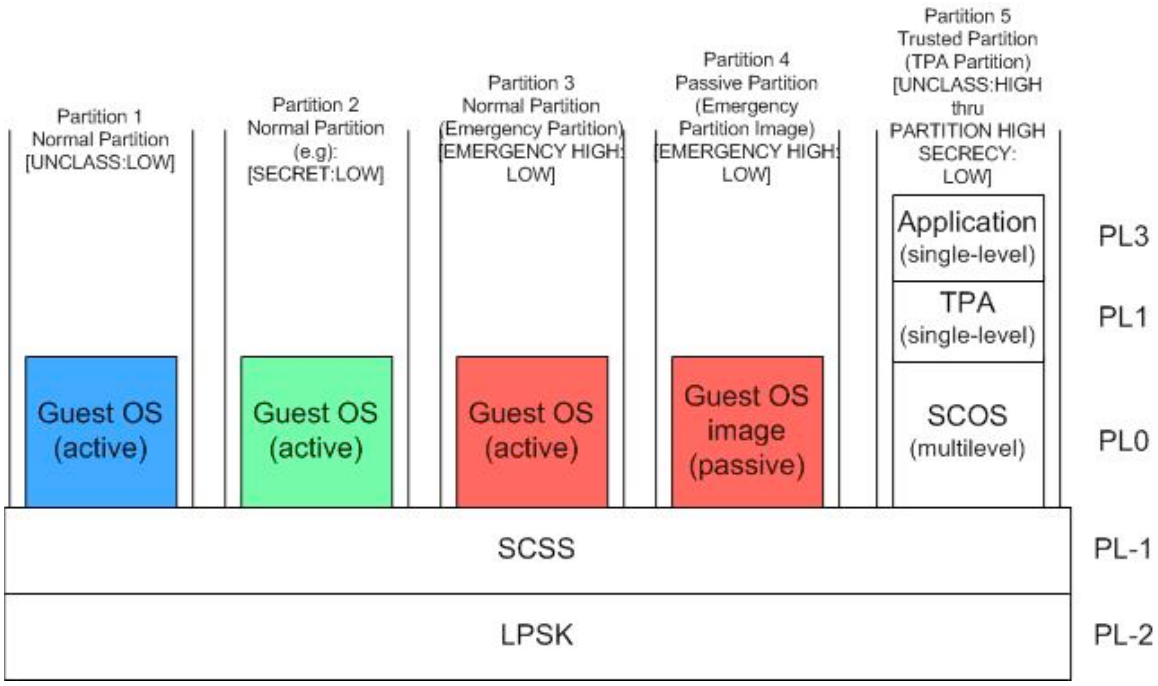

Figure 6. Phase II Architecture

A guest OS will be able to run in PL0 of low-integrity partitions. The supported guest OS will be selected at a later date. In other words, there will be only one particular OS that will be supported in this environment. Applications activated by the TPA will be run in a different PL than the TPA. The advanced power states described in Section 3.1 will also be available.

4.4 Phase III

Drafts of the high-level specifications for the features prototyped in Phase II will be completed, with the additional goal of specifying a TML that can qualify as a Type I Virtual Machine Monitor (VMM).

4.5 Phase Summary

A summary of the four phases is shown in Table 2.

		Phase 0	Phase I	Phase II	Phase III
Deliverables	Prototype	X	X	X	
	Specifications		X		X
Functionality	Transient Trust		X	X	X
	Virtualized network card		X	X	X
	Advanced Power States			X	X
	Six PLs			X	X
	Guest OS			X	X
	Handheld form factor				X
	Type I VMM				X

Table 2. Phase Summary

5 Phase I Usage

It is assumed that the Phase III SecureCore handheld will present a user experience that is more similar to that of a PC, such as the new Ultra Mobile PC (UMPC) devices, as opposed to that of a traditional Personal Digital Assistant (PDA), such as a Palm. In the traditional PDA experience, the device is initialized once (or rarely, in the event of a system crash), and then put into device standby by either pressing the power button or after a defined period with no user input (referred hereafter as *user inactivity*).

An *example* user interaction with a SecureCore handheld is given below, starting with the platform in a shutdown state:

1. The TML is initialized after the user presses the power button.
2. The SCSS attempts to boot all active partitions.
   This is a reasonable approach, but may seem at first glance like a waste of resources by booting a guest OS that may not be needed by the user. However, it must be remembered that resources are allocated to each partition during platform initialization, including memory and CPU time, so waiting to boot an OS until requested by the user is a waste of the user's time.
3. The platform configuration dictates which partition has focus immediately following platform initialization.
   The TML will not allow a non-TPA partition to have initial focus unless the partition is labeled as UNCLASS:LOW. If the initial partition with focus contains a booted guest operating system, then the user can interact with it without intervention by the TPA.
   If the initial partition with focus is the TPA partition, the user must still invoke the Secure Attention Key (SAK) to gain access to a TPA menu. This requirement to invoke the SAK before issuing TPA commands is necessary to provide an un-spoofable trusted path.
4. After the SAK is invoked, the TPA partition is given focus, and the user is presented with a limited menu by the TPA (e.g., logging into the TPA). This initial limited menu is hereafter referred to as the *unauthenticated menu*. (See Table 3).

Note that changing the partition with focus does not prevent the non-TML subjects in the partition that lost focus from continuing to execute; the partition that loses focus simply loses user I/O visibility, though the non-TML subjects are oblivious to that loss. Remember that the LPSK gives each active partition a fixed time slice for the execution of its subjects. The state of the subjects in each partition is therefore saved at the end of each execution slice, and restored just before each execution slice. Therefore, partition focus and partition scheduling are orthogonal.

5. From the TPA unauthenticated menu the user may choose to change the partition with focus to an active partition that is labeled as UNCLASS:LOW. Selecting a partition changes focus to the selected partition.
6. If the user needs to perform a trusted command or change focus to an active partition that requires TPA login (i.e., the partition label dominates UNCLASS:LOW), the SAK is invoked, which changes focus to the TPA partition and presents the TPA unauthenticated menu.
7. The user selects the "TPA Login" option from the TPA unauthenticated menu. Selecting the login option prompts the user to enter a username and password (or some other authentication mechanism). The user is always associated with a session level at any given moment, as managed by the SCOS. Before the user logs into the TPA the session level is considered UNCLASS:LOW. After a user logs in, the same session level will be associated with the user unless an optional default session level has been configured for the user, or until the user sets a different session level via the TPA interface.
8. After successfully entering the password and session level, an expanded TPA menu is presented to the user that is suitable for the current session level. This expanded menu is hereafter referred to as the *authenticated menu*. (See Table 3). For example, if the current session level is TS:LOW, then the user can (among other things) change the partition with focus to any active partition that is labeled as TS:LOW. One restriction is that the emergency partition is not available in a non-emergency situation.
9. When the handheld receives a remote signal declaring an emergency situation (at an emergency security level), then the TPA partition becomes the focus (if it was not already). At this point the user is in one of two states: a) the user has not logged into the TPA yet; or b) the user has already logged into the TPA. Depending on the state, the following will occur:

   Situation a): The emergency situation is communicated to the user, and the user is prompted to invoke the SAK (to ensure the user is not being spoofed) and log into the TPA. The user is prompted to acknowledge receipt of the information. After the acknowledgment the emergency partition becomes available for focus change.

   Situation b): The emergency situation is communicated to the user, and the user is prompted to acknowledge receipt of the information. After the acknowledgment, the emergency partition becomes available for focus change.

   The TPA (or another trusted component) is trusted to communicate the identity of the authenticated user to the remote authority.

10. The emergency partition contains an emergency application. The application allows files to be read that may be provided during an emergency.
11. When the trusted communication channels have been established between the emergency partition and emergency data providers, the remote authority and third party data providers can have confidence that the sensitive data transferred to the handheld (and stored in the emergency partition) will not be available after the emergency is over. The protocol for establishing the trusted channel, such as key exchanges, will be covered in a separate document.
12. All communications between the remote authority, third party data providers and the handheld's emergency partition take place at the emergency security level. All data received by the emergency partition is stored in the emergency partition and implicitly labeled at the emergency security level. If changes to documents are made local to the handheld, and those changes need to be available after the emergency, then they must be transferred back to the provider before the end of the emergency. The details of this protocol will be the focus of future work.
13. When a user changes focus to the TPA while the handheld is in an emergency situation, a different set of TPA menu options will be presented to the user, hereafter referred to as the *emergency menu*. (See Table 3).
14. When a signal is given from a central authority that the emergency is over, the TPA partition becomes the partition with focus, and the emergency partition is made inaccessible to the user and halted. Invoking the SAK at this point presents the authenticated menu. Exiting an emergency situation requires all the data in the emergency partition to be securely expunged to meet the transient trust requirement. In addition, the emergency partition is restored to a clean state so it will be ready for the next emergency.

6 TPA Requirements

This section describes TPA requirements for Phase I of the SecureCore handheld. In addition, a summary of the required TPA menus is provided, followed by a state diagram for the TPA.

6.1 TPA User Interaction

1. The TPA shall display prompts and other output to the user.
2. The TPA shall accept user input.
3. The TPA shall require a Secure Attention Key (SAK) be invoked before any TPA menu options are displayed.
4. If a user is not yet logged in to the TPA, the TPA shall provide limited menu options after the SAK is invoked, known as the TPA *unauthenticated menu*. (See Table 3).
5. The TPA unauthenticated menu shall provide an option for logging into the TPA.
6. The TPA shall prompt for user ID and authentication data (e.g., a password) to facilitate the login request.
7. After a user has logged in and established a session level, the TPA shall provide an expanded menu that is relevant to the current session level, known as the TPA *authenticated menu*. (See Table 3).

8. After a user is authenticated, TPA menus shall provide an option for setting the session level.
   This option allows the user to perform actions that are only possible at a particular level, such as changing focus to a different partition at a particular level. The possible session level settings are restricted by the user's clearance.
9. All TPA menus shall provide an option for displaying the current session level so the user is not required to remember the default session level or the last session level that was manually selected.
10. The TPA authenticated menu shall provide an option for changing the current user's password.
    This option is not available during an emergency because there is a high risk for users to forget passwords after they are changed, so it was determined to be a dangerous option during an emergency situation.
11. All TPA menus shall provide an option to interact with other active partitions which, when selected, results in a change to the partition with focus.
    One of the basic capabilities of the TPA is the ability to change the partition with focus, so focus changing is a potential option for all TPA states. The TPA menus shall only display those active partitions that have a security level that is dominated by the current session level. A user shall only be able to change the partition with focus to partitions that are equal to the current session level.
12. The TPA authenticated menu shall provide an option for setting the default session level for the current user.
    This is a usability feature. It allows a user to explicitly set the session level to be used after a successful TPA authentication. If set, the default session level is used instead of the implicit default security level of UNCLASS:LOW.
13. After a user is authenticated, TPA menus shall provide an option for entering the User Master Key (UMK).
    If future research determines that the UMK is **required** during emergency operations, then the "Load User Master Key (UMK)" option shall be removed from the emergency menu, and the TPA shall require the entry of the UMK during transition to the emergency menu.
14. After a user is authenticated, TPA menus shall provide an option for logging out of the TPA.
    Selecting the TPA logout option reduces the TPA menu options to the unauthenticated menu. Logging out of the TPA shall not affect other active partitions in any way.
15. All TPA menus shall provide an option to halt a partition.
    The halt option provides the user with the ability to reboot a problem partition (e.g., a hung partition). The user shall only be able to perform a partition halt for active partitions that exist at the current session level.
16. All TPA menus shall provide an option to boot a partition.
    The boot option gives the user the ability to boot partitions the user needs to interact with, but which are not currently booted. The user shall only be able to boot active partitions that exist at the current session level.
17. All TPA menus shall provide an option for performing a platform shutdown.

The shutdown option will allow the user to turn off the handheld (e.g., to conserve power), and should be allowed at any TPA state, even if the user is not currently logged in.

18. The TPA shall not maintain the state of the current session.
    The TPA shall depend on the SCOS to maintain the following state information: 1) the user's current session level, 2) the login status (whether the user is logged in or out), and 3) the emergency status.

6.2 TPA Emergency Handling

1. Upon notification of a declared emergency, the TPA shall notify the user of the emergency and prompt the user to invoke the SAK.
2. After the invocation of the SAK (described above), the TPA shall ensure the user is authenticated, shall inform the user of the nature of the emergency (as provided by the remote authority), and shall present an appropriate menu, known as the *emergency menu*. (See Table 3).
3. The emergency menu shall provide an option to redisplay the emergency description.
4. The emergency menu shall provide an option for changing focus to the emergency partition(s).

6.3 TPA Application Support

1. The TPA shall be able to start an application that is installed in the TPA partition. The start of an application is the result of a TPA menu selection from either the TPA authenticated menu or TPA emergency menu, or as a result of an event.
2. The TPA shall execute applications at the user's current session level.
3. The TPA shall provide a menu option to resume a suspended application.
4. The TPA shall provide a menu option to terminate a suspended application.

6.4 TPA Interactive Menu

Table 3 shows the minimal TPA menu for Phase I, as culled from the TPA requirements. There are three different menus, depending on whether the user has logged into the TPA, and whether there is currently an emergency in progress.

Unauthenticated Menu	Authenticated Menu	Emergency Menu
TPA Login	TPA Logout	TPA Logout
Change Focus	Change Focus	Change Focus
Platform Shutdown	Platform Shutdown	Platform Shutdown
Halt Partition	Halt Partition	Halt Partition
Boot Partition	Boot Partition	Boot Partition
	Load User Master Key (UMK)	Load User Master Key (UMK)
	Start Application	Start Application
	Resume Application	Resume Application
	Terminate Application	Terminate Application
	Set Session Level	Set Session Level
Display Session Level	Display Session Level	Display Session Level
	Set Default Session Level	Set Default Session Level
	Change Password	
		Display emergency description

Table 3. TPA Menus

6.5 TPA State Diagram

Figure 7 shows the state diagram for Phase I of the TPA, as culled from the requirements.

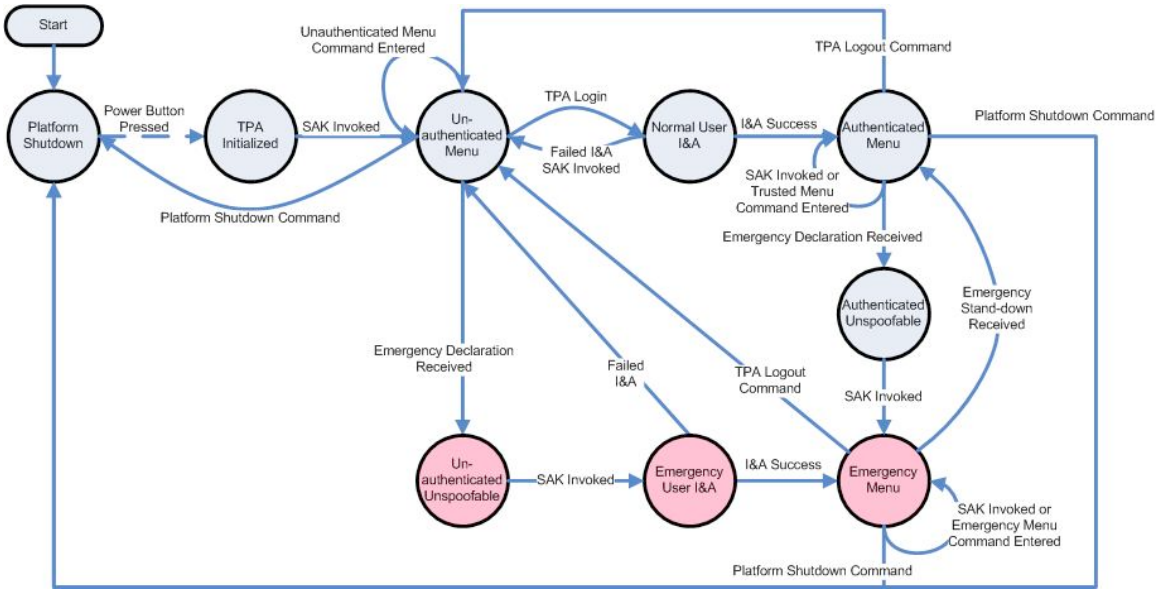

Figure 7. TPA State Diagram

7 Additional Research

Additional research is needed in many areas. The following are topics that will be researched by CISR:

1. The need for an administrative interface for making local and/or remote configuration changes to fielded handhelds must be studied. One contingency to

be included in the research and decision making is the possibility of a user forgetting the password in the field during an emergency.

2. The necessity and scope of a trusted audit mechanism needs to be determined. The inclusion of an audit mechanism may change some design decisions, such as which trusted component performs I&A.
3. With respect to "key management" the requirements currently only make reference to the User Master Key (UMK). Research is required to determine whether additional key management functionality is needed or desired.

Research is also needed on the following topics, but are not of specific interest to CISR:

1. Research is needed on various power-saving modes, such as Standby and Hibernate. Specifically, the Advanced Configuration and Power Interface (ACPI) needs to be better understood with respect to SecureCore.
2. Usability issues for first responders should be researched. Is it possible for a fireman to use a handheld device if he must be wearing thick gloves? Can a SAK be a unique key sequence if only one hand is available for user interaction? Is a stylus too awkward and prone to loss or drops to be useful in an emergency? What is a good user interface for first responders?

References

[1] Holmberg, David G., David, William D, Treado, Stephen J., Reed, Kent A., *Building Tactical Information System for Public Safety Officials: Intelligent Building Response*, NISTIR 7314, National Institute of Standard and Technology, January 2006.

[2] Benzel, Terry V., Irvine, Cynthia E., Levin, Timothy E., Bhaskara, Ganesha, Nguyen, Thuy D., Clark, Paul C., *Design Principles for Security*, NPS-CS-05-010, Naval Postgraduate School, September 2005.

[3] Common Criteria for Information Technology Security Evaluation, version 2.3, August 2005.

[4] Glossary of Computer Security Terms, NCSC-TG-004, version 1, National Computer Security Center, October 21, 1988.

Initial Distribution List

1. Defense Technical Information Center — 2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library, Code 013 — 2
   Naval Postgraduate School
   Monterey, CA 93943-5100

3. Research Office, Code 09 — 1
   Naval Postgraduate School
   Monterey, CA 93943-5138

4. Karl Levitt — 1
   National Science Foundattion
   4201 Wilson Blvd.
   Arlington, VA 22230

5. Lee Badger — 1
   DARPA
   3701 Fairfax Drive
   Arlington, VA 22203

6. Paul C. Clark — 2
   Code CS/Cp
   Department of Computer Science
   Naval Postgraduate School
   Monterey, CA 93943-5118

7. Cynthia E. Irvine — 2
   Code CS/Ic
   Department of Computer Science
   Naval Postgraduate School
   Monterey, CA 93943-5118

8. Timothy E. Levin — 2
   Code CS/Tl
   Department of Computer Science
   Naval Postgraduate School
   Monterey, CA 93943-5118

9. Thuy D. Nguyen 2
Code CS/Tn
Department of Computer Science
Naval Postgraduate School
Monterey, CA 93943-5118

10. Timothy M. Vidas 2
Code CS
Department of Computer Science
Naval Postgraduate School
Monterey, CA 93943-5118

[THIS PAGE IS INTENTIONALLY BLANK]