ACL Version 2.0.1 and eSNACC Version 1.3 Security Target
April 15, 2005
Document No. E3-1001-003 (10)

COACT, Inc.
Rivers Ninety Five
9140 Guilford Road, Suite G
Columbia, MD 21046-2587
Phone: 301-498-0150
Fax: 301-498-0855

DOCUMENT INTRODUCTION

Prepared By:

COACT, Inc.
9140 Guilford Road, Suite G
Columbia, Maryland 21046-2587

Prepared For:

Getronics Government Solutions
2525 Network Place
Herndon, VA 20171

This document provides the basis for an evaluation of a specific Target of Evaluation (TOE), the ACL version 2.0.1 and eSNACC version 1.3. This Security Target (ST) defines a set of assumptions about the aspects of the environment, a list of threats that the product intends to counter, a set of security objectives, a set of security requirements and the IT security functions provided by the TOE which meet the set of requirements.

REVISION HISTORY

Rev	Description
	November 27, 2001, initial release
1	March 1, 2002, working draft
2	September 16, 2002, updates to new ST format and corrections based on ASE evaluation
3	October 21, 2002, updates to Chapter 4.
4	November 26, 2002, updates to Chapter 2.
5	January 24, 2003 updates to Chapter 5.
6	March 20, 2003 updates to Chapters 3 and 4
7	April 23, 2003 updates to Chapters 3 and 4 for domain separation and non-bypassibility; editorial changes in all chapters.
8	April 24, 2003 updates to reflect suggested ETR changes.
9	June 17, 2003 updates to Chapter 6 Section 6.1
10	April 15, 2005 complete document usage table

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS LIST

AC	Attribute Certificate
ACDF	Access Control Decision Function
ASN	Abstract Syntax Notation
CC	Common Criteria
CML	Certificate Management Library
DER	Distinguished Encoding Rules
DLL	Dynamically Linked Library
DMS	Defense Message System
EAL3	Evaluation Assurance Level 3
IT	Information Technology
LDAP	Lightweight Directory Access Protocol
NIAP	National Information Assurance Partnership
PP	Protection Profile
PRBAC	Partition Rule Based Access Control
SF	Security Function
SFP	Security Function Policy
SOF	Strength of Function
SPIF	Security Policy Information File
ST	Security Target
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSFI	TSF Interface
TSP	TOE Security Policy

CHAPTER 1

1. Security Target Introduction

This Security Target (ST) describes the objectives, requirements and rationale for the ACL version 2.0.1 and eSNACC version 1.3. The language used in this Security Target is consistent with the *Common Criteria for Information Technology Security Evaluation, Version 2.1*, the ISO/IEC JTC 1/SC27, *Guide for the Production of PPs and STs, Version 0.9* and all National Information Assurance Partnership (NIAP) interpretations through December 20, 2001. As such, the spelling of terms is presented using the internationally accepted English.

1.1 Security Target Reference

This section provides identifying information for the ACL and eSNACC Security Target by defining the Target of Evaluation (TOE).

1.1.1 Security Target Name

ACL Version 2.0.1 and eSNACC Version 1.3 Security Target, revision 10, dated April 15, 2005.

1.1.2 TOE Reference

Access Control Library (ACL) version 2.0.1 and Enhanced Sample Neufeld ASN.1 to C/C++ Compiler (eSNACC) version 1.3.

1.1.3 Security Target Evaluation Status

This ST has been evaluated.

1.1.4 Evaluation Assurance Level

Functional claims are extended. Assurance claims conform to EAL3 (Evaluation Assurance Level 3) from the *Common Criteria for Information Technology Security Evaluation, Version 2.1* and includes the following augmentations: Subset of Implementation of the TSF, ADV_IMP.1; Descriptive Low-level design, ADV_LLD.1; Developer Defined Life Cycle Model, ALC_LCD.1; and Well-defined Development Tools, ALC_TAT.1.

1.1.5 Keywords

Clearance attributes, security labels, X.509 certificate, access control, security policy.

1.2 TOE Overview

This Security Target defines the requirements for the ACL version 2.0.1 and eSNACC version 1.3. The TOE is comprised of two software libraries that supply the IT-environment with a value needed to perform access control decisions based on X.509 certificates. The ACL portion of the TOE is composed of a high level library that performs an access control decision function. The ACL provides an Access Control Decision Function that determines if a subject's authorizations allow the subject to

access data labeled with specific sensitivity values. The ACL uses the Enhanced Sample Neufeld ASN.1 to C/C++ Compiler (eSNACC) portion of the TOE to perform decoding of certificates. eSNACC decodes X.509 Certificates, Certificate Revocation Lists and Attribute Certificates. To ensure that authorisations are commensurate with values in a security label, the ACL uses Security Policy Information Files (SPIFs). A SPIF is composed of a list of available authorizations and sensitivities along with their human readable bitmapped integer representations. By using SPIFs, the ACL can support a variety of security policies and equivalency mappings between security policy values. The ACL checks a security label to ensure it includes a valid combination of security classification and security category values as specified in the SPIF.

1.2 Security Target Organisation

Chapter 1 of this ST provides introductory and identifying information for the TOE.

Chapter 2 describes the TOE and provides some guidance on its use.

Chapter 3 provides a security environment description in terms of assumptions, threats and organisational security policies.

Chapter 4 identifies the security objectives of the TOE and of the Information Technology (IT) environment.

Chapter 5 provides the TOE security and functional requirements, as well as requirements on the IT environment.

Chapter 6 is the TOE Summary Specification, a description of the functions provided by the ACL to satisfy the security functional and assurance requirements.

Chapter 7 identifies claims of conformance to a registered Protection Profile (PP).

Chapter 8 provides a rationale for the security objectives, requirements, TOE summary specification and PP claims.

1.3 Common Criteria Conformance

The ACL version 2.0.1 and eSNACC version 1.3 is compliant with the Common Criteria (CC) Version 2.1, and assurance requirements (Part 3) augmented for EAL3: ADV_IMP.1, ADV_LLD.1, ALC_LCD.1, and ALC_TAT.1.

1.4 Protection Profile Conformance

This ST does not claim conformance to any registered Protection Profile.

CHAPTER 2

2. TOE Description

This section provides the context for the TOE evaluation by identifying the product type and describing the evaluated configuration.

2.1 Access Control Library and eSNACC Library TOE Description

The Access Control Library (ACL) and Enhanced Sample Neufeld ASN.1 to C/C++ Compiler (eSNACC) Library are modules of the freeware security libraries developed by Getronics Government Solutions. The freeware security libraries include the eSNACC Compiler/Library, S/MIME Freeware Library, Certificate Management Library, and Access Control Library. Only the ACL and eSNACC Libraries are in the TOE. Each library is independently compiled and creates a final Dynamically Linked Library (DLL) file. The ACL DLL file is named acl.dll and the eSNACC DLL file is named cpppasn1.dll.

The Access Control Library is designed using object-oriented techniques. The ACL portion of the TOE provides a function that supplies the calling application (IT-environment) a value needed to perform an access control decision. This function is commonly referred to as the Access Control Decision Function (ACDF – the TOE Security Function) that determines if a subject's[1] authorizations (contained in a X.509 Clearance attribute) allow the subject to access data labeled with specific sensitivity values (included in a security label). The ACL uses the eSNACC library to extract a clearance attribute from the X.509 certificate. The ACL then compares the security label with the security policy defined in the Security Policy Information File (SPIF). If the security label meets the security policy requirements, the ACL returns a logical true. Otherwise it returns a logical false. The ACL uses a C++ language Application Programming Interface (API) that provides the application (e.g. email, web browser/server) an interface to the ACDF.

The ACL uses the eSNACC portion of the TOE to perform decoding of certificates. The eSNACC encodes and decodes objects, such as X.509 Certificates, Certificate Revocation Lists,Security Lables, and Security Policy Information Files.

To ensure that authorisations are commensurate with values in a security label, the ACL uses SPIFs. A SPIF is composed of a list of available authorizations and sensitivities along with their human readable bitmapped integer representations supplied by the application. By using SPIFs, the ACL can support a variety of security policies and equivalency mappings between security policy values. The

[1] For the remainder of this ST, the reference to subject is not an actual subject but a representation of the subject (in this case the authorizations contained in the X.509 certificate).

ACL checks a security label to ensure it includes a valid combination of security classification and security category values as specified in the SPIF.

The intended use of the ACL is to meet the Partition Rule Based Access Control (PRBAC) processing requirements specified in the SDN.801 MISSI Access Control Concept and Mechanisms document. In addition, the ACL processes an X.509 Attribute Certificate (AC) or Version 3 X.509 public key certificate to extract a subject's Clearance attribute(s).

In summary the ACL provides an Access Control Decision Function as defined in the SDN.801 Partition Rule Based Access Control requirements using: Clearance attribute(s) containing a subject's authorizations; security label indicating sensitivity of data; and SPIF. It checks a security label to ensure that it includes a valid combination of security classification and category values as specified in the SPIF for the security policy identified in the security label.

2.1.1 Physical Boundary

Given the nature that the TOE is made up of two software libraries, the physical boundary is the compiled DLLs. Each software library is independently compiled into a DLL file. When compiled into object code, the ACL DLL file is named acl.dll and the eSNACC DLL file is named cpppasn1.dll. The boundary around these two DLL files is the physical boundary as shown in the diagram below. The interfaces to each of the libraries are the Application Programmers Interface (API) calls.

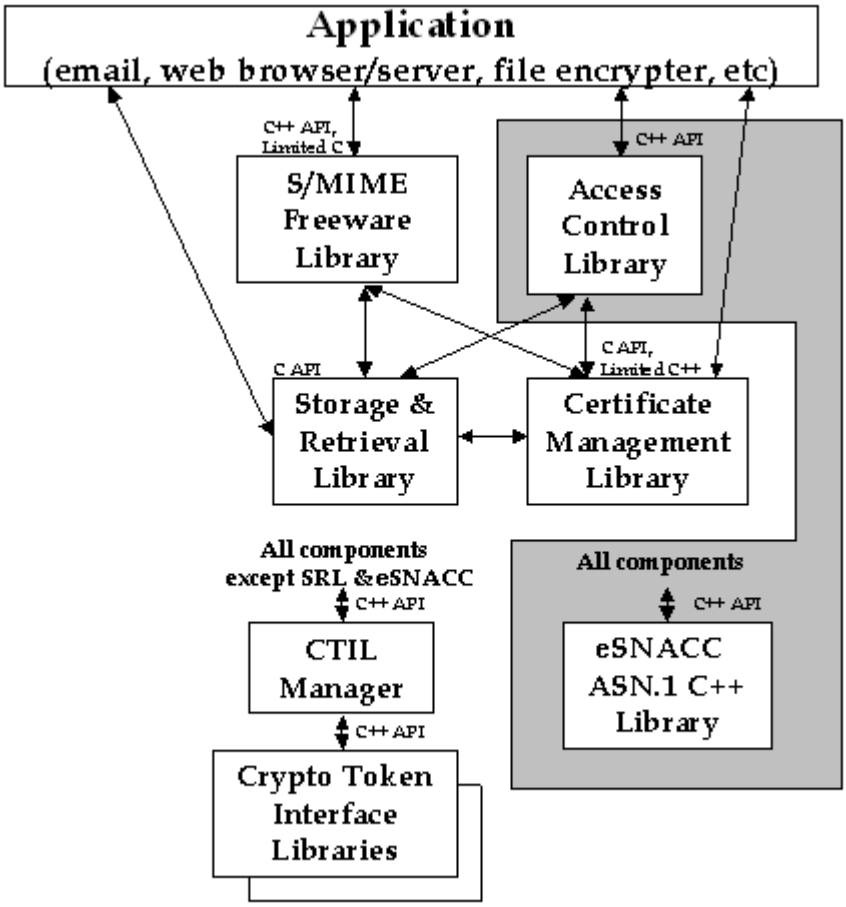

Figure 1 - TOE Boundary

2.1.2 Logical Boundary

The logical boundary of the TOE is two software libraries that are independently compiled, the ACL library that is compiled into the acl.dll file and the eSNACC ASN.1 library that is compiled into the cppasn1.dll file.

2.2 TOE Evaluated Configuration

The TOE is comprised of two libraries. The evaluation of the TOE covers only the TOE's Security Functionality. All other uses of the TOE (to meet SDN.801 and other standards) are outside of the scope of the evaluation. All other software libraries besides the ACL and eSNACC are also outside of the scope of the evaluation.

The evaluated configuration of the TOE must be executed on a Windows 98/NT/2002/XP system compiled with Microsoft Visual C++ 6.0 with service pack 4. The calling application (application that makes API calls to the ACL) is assumed to enforce the access control policy as the ACL only provides a

function to assist in the access control decision (defined in the SPIF). The evaluated configuration of the ACL must be used in the following manner:

1) The use of Certificate Management Library (CML) to validate CertificationPaths must be disabled.
2) The application is responsible for CertificationPath validation and signature validation of Attribute Certificate, Security Policy Information Files (SPIF), and Certificates.
3) The ACL must not be used to perform equivalency mapping.
4) The ACL must not be used to perform Lightweight Directory Access Protocol (LDAP) lookups.
5) The ACL source code and binary libraries must be obtained via CD/ROM media.

In order to meet these restrictions, the ACL session object must be configured as follows:

1) Create an ACL session object.
2) Pass “true” to the Session::disableValidation() method.
3) Pass “false” to the Session::enableEquivalencies() method.
4) Do not use Session::enableCML() method.
5) Do not use Session::enableLDAP() method.

CHAPTER 3

3. Security Environment

3.1 Introduction

This chapter identifies the following:

A) Significant assumptions about the TOE's operational environment.

B) IT related threats to the organisation countered by the TOE.

C) Environmental threats requiring controls to provide sufficient protection.

D) Organisational security policies for the TOE as appropriate.

Using the above listing, this chapter identifies assumptions (A), threats (T) and organisational security policies (P). For assumptions, threats or policies that apply to the environment, the initial character is followed by a period and then an 'E'. For example, A.E.PHYSICAL is a security environmental assumption of physical access protection.

3.2 Assumptions

The specific conditions listed in the following subsections are assumed to exist in the TOE environment. These assumptions include both practical realities in the development of the TOE security requirements and the essential environmental conditions on the use of the TOE.

3.2.1 Personnel Assumptions

A.E.ADMIN Administrators of the TOE are assumed to be responsible, non-evil individuals who will correctly install the TOE on the correct platform, in a manner consistent with site-specific security requirements.

3.2.2 Connectivity Assumptions

A.E.PROP Applications or programs that use the TOE will correctly interpret the results returned by the ACDF and implement the access control decision accordingly.

A.E.INT The TOE will be integrated in an application and administered by users who are responsible.

A.E.INPUT SPIFs, X.509 certificates, and security labels, contain the valid authorizations and assertions.

A.E.PROTECT The TOE will be installed on a system that provides domain separation and non-bypassibility of the security function provided by the TOE.

3.2.3 Physical Assumptions

A.E.PHYSICAL The TOE will be installed on a system that is physically protected according to site-specific requirements.

3.3 Threats

3.3.1 Threats Addressed by the TOE and the Environment

T.DECISION An unauthorized subject may gain access to objects protected by the IT-environment due to failure of the system (TOE & IT-environment) to restrict access.

3.4 Organisational Security Policies

There are no Organisational Security Policies.

CHAPTER 4

4. Security Objectives

4.1 Security Objectives for the TOE

All of the objectives listed in this section ensure that all of the security threats addressed by the TOE and the Environment listed in Chapter 3 have been countered. The security objectives (O) for Access Control Library are:

O.DECIDE The TOE will provide an access control decision function based on security attributes and a security policy.

4.2 Security Objectives for the IT Environment

O.E.PROTECT The Administrator will ensure that the IT environment provides domain separation and non-bypassibility of the security function provided by the TOE.

O.E.ACCESS The IT environment uses the TOE's ACDF to enforce access control decisions.

4.3 Security Objectives for the Non-IT Environment

O.E.ADMIN Those responsible for the TOE must ensure that the TOE is properly installed and configured on the system to meet the evaluated configuration requirements and in a manner consistent with site-specific security requirements.

O.E.INT The TOE will be integrated in an application and administered by users who are responsible.

O.E.INPUT All inputs to the TOE contain correct security labels and authorizations.

4.4 Security Objectives Rationale

Table 1 demonstrates the correspondence between the security objectives listed in Sections 4.1 and 4.2 to the assumptions, threats and policies identified in Sections 3.2, 3.3 and 3.4.

Table 1 - Correspondence Between Assumptions, Threats and Policies to Objectives

Table Legend		
A = Assumption, P = Policy, T = Threat, O = Objective, .E = Environment		
Assumption, Threat or Policy	**Security Objectives**	**Rationale**
T.DECISION - An unauthorized subject may gain access to objects protected by the IT-environment due to failure of the system (TOE & IT-environment) to restrict access.	O.DECIDE - The TOE will provide an access control decision function based on security attributes and a security policy. O.E.ACCESS - The IT environment uses the TOE's ACDF to enforce access control decisions.	The ACL will provide an Access Control Decision Function that supports access control decisions that are used by the IT environment to prevent unauthorised users from gaining access to objects where security attributes do not match. It is incumbent upon the IT environment to correctly

Table Legend		
A = Assumption, P = Policy, T = Threat, O = Objective, .E = Environment		
Assumption, Threat or Policy	**Security Objectives**	**Rationale**
		interpret and implement the ACL decision function and supply the ACL with the correct security labels and authorizations as input into the decision function process.
A.E.ADMIN - Administrators of the TOE are assumed to be responsible, non-evil individuals who will correctly install the TOE on the correct platform, in a manner consistent with site-specific security requirements.	O.E.ADMIN - Those responsible for the TOE must ensure that the TOE is properly installed and configured on the system to meet the evaluated configuration requirements and in a manner consistent with site-specific security requirements.	The administrator is responsible, trusted, and will install the TOE correctly on a system that meets the evaluated configuration requirements and in a manner consistent with site-specific security requirements to ensure correct operation of the TOE and to ensure the physical protection of the TOE.
A.E.PROP - Applications or programs that use the TOE will correctly interpret the results returned by the ACDF and implement the access control decision accordingly.	O.E.ACCESS - The IT environment uses the TOE's ACDF to enforce access control decisions.	The TOE provides a decision function only. The applications calling the TOE will correctly interpret and implement the TOE decision function.
A.E.INT - The TOE will be integrated in an application and administered by users who are responsible.	O.E.INT - The TOE will be integrated in an application and administered by users who are responsible.	Those responsible for the TOE will integrate and use the TOE in a responsible manner consistent with the evaluated configuration.
A.E.INPUT - SPIFs, X.509 certificates, security labels contain valid authorizations, and assertions.	O.E.INPUT - All inputs to the TOE contain correct security labels and authorizations.	It is incumbent upon the environment to supply the TOE the correct security labels and authorizations necessary for the TOE to arrive at the proper decision. If the environment fails to supply the correct security labels and authorizations the TOE, while performing correctly and providing the environment with the correct value for the decision, may

Table Legend		
A = Assumption, P = Policy, T = Threat, O = Objective, .E = Environment		
Assumption, Threat or Policy	**Security Objectives**	**Rationale**
		arrive at an improper decision.[2]
A.E.PROTECT - The TOE will be installed on a system that provides domain separation and non-bypassibility of the security function provided by the TOE.	O.E.PROTECT - The Administrator will ensure that the IT environment provides domain separation and non-bypassibility of the security function provided by the TOE.	The administrator is responsible, trusted, and will install the TOE in an IT environment that provides domain separation and non-bypassibility of the security function of the TOE.
A.E.PHYSICAL - The TOE will be installed on a system that is physically protected according to site-specific requirements.	O.E.ADMIN - Those responsible for the TOE must ensure that the TOE is properly installed and configured on the system to meet the evaluated configuration requirements and in a manner consistent with site-specific security requirements.	The administrator is responsible, trusted, and will install the TOE correctly on a system that meets the evaluated configuration requirements and in a manner consistent with site-specific security requirements to ensure correct operation of the TOE and to ensure the physical protection of the TOE.

[2] ST author's note: 'Correct' meaning computationally correct: 'improper' meaning not correct from a security authorization viewpoint. GIGO.

CHAPTER 5

5. IT Security Requirement

This section contains the functional requirements that are provided by the TOE and IT environment. These requirements include two explicitly stated functions.

5.1 Security Functional Requirement

Table 2 lists the functional requirement and the security objective the requirement enforces. All functional and assurance dependencies associated with the components in Table 2 have been satisfied.

Table 2 - Security Functional Components

CC Component	Name	Hierarchical To	Dependency	Objectives Enforced / Rationale
FXP_ACD.1	Access Control Decisions	No Other Components	None	O.E.ACCESS
FXP_ACF.1	Access Control Decision Function	No Other Components	None	O.DECIDE
FPT_RVM.1	Non-bypassability of the TSP	No Other Components	None	O.E.PROTECT
FPT_SEP.1	TSF domain separation	No Other Components	None	O.E.PROTECT

The functional requirements that appear above are described in more detail in the following subsections. These bracketed items include either “assignments” that are TOE specific or “selections” from the Common Criteria that the TOE enforces.

5.1.1 TOE Security Functional Requirement

Justification: The ACL does not enforce access control. It provides a decision function for access control to be implemented. Therefore this requirement is explicitly stated.

5.1.1.1 FXP_ACF.1 Access Control Decision Function

Hierarchical to: No Other components

FXP_ACF.1.1 The TSF shall provide a decision function on the [assignment: *Security Policy defined in the SPIF*] and [assignment: *X.509 certificates*] to supply the IT-environment with a value that access control decisions can be made.

Rationale: Only those users with assigned valid clearance attributes will be granted access to information labeled with specific sensitivity values. An Access Control Decision Function (ACDF) determines if a subject’s authorisations, contained in an X.509 clearance attribute allow access to objects with specific sensitivity values that are included in a security label. The SPIF is

used as part of the process of ensuring that the subject's authorisations correspond with the values of the security label. This function enforces the TOE objective, O.DECIDE.

5.1.2 Security Functional Requirements Levied on the IT environment

Justification: The ACL does not enforce access control. It provides a decision function for access control to be implemented by the IT-environment. The environment uses the ACDF to make access control decisions. Therefore this requirement is explicitly stated.

5.1.2.1 FXP_ACD.1 Access Control Decision

Hierarchical to: No Other components

FXP_ACD.1.1 The IT Environment shall make a decision and enforce the [assignment: *Security Policy defined in the SPIF*] based on [assignment: *X.509 certificates*] by using the TOE ACDF to enforce the Security Policy.

Rationale: Only those users with assigned valid clearance attributes will be granted access to information labeled with specific sensitivity values. An Access Control Decision Function (ACDF) determines if a subject's authorisations, contained in an X.509 clearance attribute allow access to objects with specific sensitivity values that are included in a security label. The SPIF is used as part of the process of ensuring that the subject's authorisations correspond with the values of the security label. This SFR enforces the environmental objective, O.E.ACCESS.

5.1.2.2 FPT_RVM.1 Non-Bypassability of the TSP

Hierarchical to: No other components.

Refinement (in bold):

FPT_RVM.1.1 The **IT-environment** shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

Dependencies: No dependencies.

Rationale: The IT Environment must ensure that the TOE is protected from being bypassed. Along with the Operating System, the calling application using the TOE must ensure the TOE's ACDF is used and not bypassed. Along with SFR FPT_SEP.1, this requirement enforces the Environmental objective, O.E.PROTECT.

5.1.2.3 FPT_SEP.1 TSF Domain Separation

Hierarchical to: No other components.

Refinement (in bold):

FPT_SEP.1 The **IT-environment** shall maintain a security domain for **TOE** execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.1.2 The **IT-environment** shall enforce separation between the security domains of subjects in the **IT-environment Scope of Control**.

Dependencies: No dependencies.

Rationale: The IT Environment, specifically the Operating System, must supply the TOE with a secure domain for its execution, protecting it from corruption. Along with SFR FPT_RVM.1, this requirement enforces the Environmental objective, O.E.PROTECT.

5.2 TOE Security Assurance Requirements

The TOE meets the assurance requirements for EAL3 augmented. These requirements are summarised in Table 3.

Table 3 - Assurance Requirements

Assurance Class	Component ID	Component Title
Configuration Management	ACM_CAP.3	Authorisation Controls
Configuration Management	ACM_SCP.1	TOE CM Coverage
Delivery and Operation	ADO_DEL.1	Delivery Procedures
Delivery and Operation	ADO_IGS.1	Installation, Generation, and Start-Up Procedures
Development	ADV_FSP.1	Informal Functional Specification
Development	ADV_HLD.2	Security enforcing high-level design
Development	ADV_IMP.1	Subset of the Implementation of the TSF
Development	ADV_LLD.1	Descriptive Low-Level Design
Development	ADV_RCR.1	Informal Correspondence Demonstration
Guidance Documents	AGD_ADM.1	Administrator Guidance
Guidance Documents	AGD_USR.1	User Guidance
Life Cycle Support	ALC_DVS.1	Identification of Security Measures
Life Cycle Support	ALC_LCD.1	Developer Defined Life-Cycle Model
Life Cycle Support	ALC_TAT.1	Well-defined Development Tools
Tests	ATE_COV.2	Analysis of coverage
Tests	ATE_DPT.1	Testing High-Level Design
Tests	ATE_FUN.1	Functional Testing
Tests	ATE_IND.2	Independent Testing - Sample
Vulnerability Assessment	AVA_MSU.1	Examination of Guidance
Vulnerability Assessment	AVA_SOF.1	Strength of TOE Security

Assurance Class	Component ID	Component Title
		Function Evaluation
Vulnerability Assessment	AVA_VLA.1	Developer Vulnerability Analysis

CHAPTER 6

6. TOE Summary Specification

6.1 TOE Security Functions

The major functions implemented by the TOE are:

PROCESS The TOE processes X.509 certificates, Security Labels and SPIFs to indicate the success or failure of an access control decision to the IT environment. Through API calls, the ACL accepts an X.509 certificate, Security Label and Security Policy Information File (SPIF). The Security Label is compared with the SPIF to determine if it contains a valid combination of values. The ACL then performs a validity check of the X.509 certificate and extracts the clearance attribute from the X.509 Subject Directory Attributes extension. The ACL then performs the ACDF using the clearance attribute, Security Label, and SPIF. If the clearance attribute meets the requirements of the security policy no error is indicated. Otherwise an error is indicated.

Table 4 - Functions to Security Functional Requirements Mapping

Functions	Security Functional Requirements	Rationale
PROCESS	FXP_ACF.1	The PROCESS function performs the access control decision function based on a Security Policy Information File, Security Label, and Clearance attribute extracted from an X.509 certificate.

Table 5 shows the mapping between the security functional requirements and the functions listed above.

Table 5 - Security Functional Requirements to Functions Mapping

Security Functional Requirement	Functions	Rationale
FXP_ACF.1	PROCESS	The PROCESS Security Function makes the Access Control Decision Function. The ACL compares an extracted clearance attribute, to a Security Policy Information File and Security Label to make the access control decision.

6.2 Assurance Measures

The assurance measures provided by the TOE satisfy all of the assurance requirements listed in Chapter 5, Table 3. Table 6 provides a reference between each TOE assurance requirement and the related vendor documentation that satisfies each requirement.

Table 6 - Assurance Measures

Assurance Component	Documentation Satisfying Component	Rationale
ACM_CAP.3	ACL Configuration Management Plan v FS01-199-00	The Configuration Management plan identifies the management of the TOE versions.
ACM_SCP.1	ACL Configuration Management Plan v FS01-199-00	The Configuration Management plan identifies the management of the TOE versions.
ADO_DEL.1	The ACL Fact Sheet, the SMP Component Setup Manual v2.0.1, CC EAL3 Supplement v2.0.1	The Fact Sheet and CC EAL3 Supplement describe the delivery of the TOE in a secure manner.
ADO_IGS.1	SMP Component Setup Manual v2.0.1	This document contains the information on the installation and start-up of the TOE.
ADV_FSP.1	ACL Application Programming Interface (API) v2.0.1, and ACL CC EAL3 Supplement v2.0.1	The API document identifies the interfaces to the TOE.
ADV_HLD.2	ACL Application Programming Interface (API) v2.0.1, ACL Software Design Description (SDD) v2.0.1, ACL CC EAL3 Supplement v2.0.1	The CC EAL3 Supplement breaks down the TOE into Groups, or Subsystems. The API and SDD documents describe the interfaces and functional process of the ACL.
ADV_IMP.1	ACL Source Code v2.0.1	The entire source code or implementation of the ACL is available.
ADV_LLD.1	ACL Application Programming Interface (API) v2.0.1, ACL Software Design Description (SDD) v2.0.1, ACL CC EAL3 Supplement v2.0.1.	The API document breaks down the ACL into classes or modules. The API also lists the interfaces of those modules. The CC EAL3 Supplement identifies the TSF subsystem.
ADV_RCR.1	ACL CC EAL3 Supplement v2.0.1	Contains a correspondence between the TSS to FSP to HLD to LLD to IMP.

Assurance Component	Documentation Satisfying Component	Rationale
AGD_ADM.1	N/A	The only administrative function of the ACL is the secure installation of the ACL which is covered in ADO_IGS.1
AGD_USR.1	ACL Application Programming Interface (API) v2.0.1	The ACL API document describes the interfaces available to the programmer and IT-environment, which combined form the user of the ACL. The IT-environment is the user at the time of execution, and a programmer is the user at the development stage of the IT-environment.
ALC_DVS.1	ACL Life-Cycle Support Plan rev. 03/07/02	The Life-Cycle Plan describes the Development site and procedures used in developing the ACL.
ALC_LCD.1	ACL Life-Cycle Support Plan rev. 03/07/02.	The developer's life-cycle plan is described in this document.
ALC_TAT.1	ACL Life-Cycle Support Plan rev. 03/07/02, CC EAL3 Supplement and ACL Source Code v2.0.1.	These two documents describe the well-defined tools used to develop the TOE. The source code shows the unambiguous language used to design the TOE.
ATE_COV.2	ACL Test Coverage Analysis Version 2.0.1, Revision 1, 29 June 2004	This Test Coverage Analysis provides an analysis of the test coverage as required for EAL3 assurance. The analysis of the test coverage l demonstrates the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification. The analysis of the test coverage demonstrates that the correspondence between the TSF as described in the functional specification and

Assurance Component	Documentation Satisfying Component	Rationale
		the tests identified in the test documentation is complete
ATE_DPT.1	ACL Test Depth Analysis, Version 2.0.1, 20 May 2004.	This Test Depth Analysis provides an analysis of the depth of testing as required for EAL3 assurance. The depth analysis demonstrates the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design.
ATE_FUN.1	Access Control Library Application Programming Interface, October 31, 2001, version 2.0; Access Control Library Software Design Description, October 31, 2001, version 2.0; Access Control Library EAL 3 Software Test Report (ACL STR), June 20, 2003, version 2.0.1; Access Control Library EAL 3 Software Test Description (ACL STD), June 20, 2003, version 2.0.1.	These documents describe that the developer's functional test documentation is sufficient to demonstrate that security functions perform as specified.
ATE_IND.2	ACL Version 2.0.1 and eSNACC Version 1.3 Security Target, June 17, 2003, Revision 9; Access Control Library Application Programming Interface, October 31, 2001, version 2.0; Access Control Library Software Design Description, October 31, 2001, version 2.0; Access Control Library EAL 3 Software Test Report, June 20, 2003, version 2.0.1; Access Control Library EAL 3 Software Test Description, June 20, 2003, version 2.0.1.	These documents were used as the premise for the independent testing to demonstrate that security functions perform as specified.
AVA_MSU.1	ACL Version 2.0.1 and eSNACC Version 1.3 Security Target, June 17, 2003, Revision 9; ACL Application Programming Interface v2.0.1;	These documents, along with the TOE, were used to determine whether the guidance is misleading, unreasonable or conflicting, whether secure procedures

Assurance Component	Documentation Satisfying Component	Rationale
	ACL Software Design Description v2.0, October 31, 2001; ACL Common Criteria EAL 3 Supplement v2.0.1; ACL Vulnerabilities and Misuse Document version 2.0.1 revision 2; Test documentation; TOE suitable for testing.	for all modes of operation have been addressed, and whether use of the guidance will facilitate prevention and detection of insecure TOE states.
AVA_SOF.1	ACL Version 2.0.1 and eSNACC Version 1.3 Security Target, June 17, 2003, Revision 9.	The ST does not identify any specific security mechanisms for which there is a SOF claim.
AVA_VLA.1	ACL Vulnerabilities and Misuse Document v2.0.1, June 4, 2004.	This document was used to perform the vulnerability analysis.

6.2.1 Rationale for TOE Assurance Requirements

The TOE stresses assurance through vendor actions that are within the bounds of current best commercial practice. The TOE provides, primarily via review of vendor-supplied evidence, independent confirmation that these actions have been competently performed.

The general level of assurance for the TOE is:

A) Consistent with current best commercial practice for IT development and provides a product that is competitive against non-evaluated products with respect to functionality, performance, cost, and time-to-market.

B) The TOE assurance also meets current constraints on widespread acceptance, by expressing its claims against EAL3 augmented from part 3 of the Common Criteria.

CHAPTER 7

7. Protection Profile Claims

This chapter provides detailed information in reference to the Protection Profile conformance identification that appears in Chapter 1, Section 1.4 Protection Profile Conformance.

7.1 Protection Profile Reference

This Security Target does not claim conformance to any registered Protection Profile.

7.2 Protection Profile Refinements

This Security Target does not claim conformance to any registered Protection Profile.

7.3 Protection Profile Additions

This Security Target does not claim conformance to any registered Protection Profile.

7.4 Protection Profile Rationale

This Security Target does not claim conformance to any registered Protection Profile.

CHAPTER 8

8. Rationale

The IT Security Objectives are met through Security Functional and Assurance requirements as a mutually supportive whole.

8.1 Security Objectives Rationale

The rationale for the security objectives of the TOE and IT-environment are defined in Chapter 4 Security Objectives Rationale.

8.2 Security Requirements Rationale

The rationale for the security requirements of the TOE and IT-environment are defined in two sections. Rationale for the security functional requirements is given after each functional component description in Chapter 5, Security Functional Requirements. Rationale for the security assurance requirements is given in Chapter 6, Rationale for the TOE Assurance Requirements.

8.3 TOE Summary Specification Rationale

The rationale for the TOE Summary Specification is defined in Chapter 6, TOE Security Functions.

8.4 PP Claims Rationale

The rationale for the Protection Profile conformance claims is defined in Chapter 7, Protection Profile Rationale.