National Information Assurance Partnership
Common Criteria Evaluation and Validation Scheme

Common Criteria Evaluation and Validation Scheme
Validation Report

BAE Systems

Access Control Library (ACL) Version 2.0.1 and eSNACC Version 1.3

Report Number: CCEVS-VR-05-0083

Dated: 22 April 2005

National Institute of Standards and Technology
Information Technology Laboratory
100 Bureau Drive
Gaithersburg, MD 20899

National Security Agency
Information Assurance Directorate
9800 Savage Road STE 6740
Fort George G. Meade, MD 20755-6740

ACKNOWLEDGEMENTS

Validation Team

Dr. Jerome Myers
The Aerospace Corporation
Columbia, Maryland

Margaret Webster-Butler
National Security Agency
Linthicum, Maryland

Common Criteria Testing Laboratory
COACT CAFÉ Laboratory
Columbia, Maryland 21046-2587

Table of Contents

Table of Figures

1 EXECUTIVE SUMMARY

This report documents the NIAP Validators' assessment of the CCEVS evaluation of BAE Systems Access Control Library (ACL) Version 2.0.1 and eSNACC Version 1.3 at EAL3 augmented with ADV_IMP.1, Subset Implementation of the TSF, ADV_LLD.1. Descriptive Low-level Design, ALC_LCD.1, Developer Defined Life Cycle Model, and

ALC_TAT.1, Well-defined Development Tools. It presents the evaluation results, their justifications, and the conformance result.

The evaluation was performed by the CAFE Laboratory of COACT Incorporated, located in Columbia, Maryland. The bulk of the evaluation was completed on 12 December 2004, but some minor documentation and delivery procedure changes were required that resulted in an official completion date of 21 April 2005. The information in this report is largely derived from the Evaluation Technical Report (ETR) written by COACT and submitted to the Validators. The evaluation determined the product conforms to the CC Version 2.1, Part 2 extended and Part 3 to meet the requirements of Evaluation Assurance Level (EAL) 3 augmented with ADV_IMP.1, Subset Implementation of the TSF, ADV_LLD.1. Descriptive Low-level Design, ALC_LCD.1, Developer Defined Life Cycle Model, and ALC_TAT.1, Well-defined Development Tools, resulting in a “pass” in accordance with CC Part 1 paragraph 175.

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 (ACDF) 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 authorizations 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.

2 Identification

The CCEVS is a joint National Security Agency (NSA) and National Institute of Standards and Technology (NIST) effort to establish commercial facilities to perform trusted product evaluations. Under this program, security evaluations are conducted by commercial testing laboratories called Common Criteria Testing Laboratories (CCTLs) using the Common Evaluation Methodology (CEM) for Evaluation Assurance Level (EAL) 1 through EAL 4 in accordance with National Voluntary Laboratory Assessment Program (NVLAP) accreditation.

The NIAP Validation Body assigns Validators to monitor the CCTLs to ensure quality and consistency across evaluations. Developers of information technology products desire a security evaluation contract with a CCTL and pay a fee for their product's evaluation. Upon successful completion of the evaluation, the product is added to NIAP CCEVS' Validated Products List. Table 1 provides information needed to completely identify the product, including:

- the Target of Evaluation (TOE): the fully qualified identifier of the product as evaluated,
- the Security Target (ST), describing the security features, claims, and assurances of the product,
- the conformance result of the evaluation,
- the organizations and individuals participating in the evaluation.

Table 1: Evaluation Identifiers

Evaluation Identifiers for BAE Systems ACL Version 2.0.1 and eSNACC Version 1.3	
Evaluation Scheme	United States NIAP Common Criteria Evaluation and Validation Scheme
TOE	ACL Version 2.0.1 and eSNACC Version 1.3
Protection Profile	N/A
Security Target	ACL Version 2.0.1 and eSNACC Version 1.3 Security Target, Revision 10, dated April 15, 2005
Evaluation Technical Report	ACL Version 2.0.1 and eSNACC Version 1.3 Evaluation Technical Report , Document No. F3-0105-006(2), Dated April 25, 2005
Conformance Result	Part 2 extended, Part 3 conformant, and EAL3 augmented with ADV_IMP.1, Subset Implementation of the TSF, ADV_LLD.1. Descriptive Low-level Design, ALC_LCD.1, Developer Defined Life Cycle Model, and ALC_TAT.1, Well-defined Development Tools,
Version of CC	CC Version 2.1 [1], [2], [3], [4] and all applicable NIAP CCEVS and International Interpretations effective on 20 December 2001

Evaluation Identifiers for BAE Systems ACL Version 2.0.1 and eSNACC Version 1.3	
Version of CEM	CEM Version 1.0 [5], [6], and all applicable International Interpretations effective on 20 December 2001
Sponsor	BAE Systems 2525 Network Place Herndon, VA 20171
Developer	BAE Systems 2525 Network Place Herndon, VA 20171
Evaluator(s)	**COACT Incorporated** Bob Roland Thomas Fisher Will Knight Deborah Causebrook Anthony Busciglio
Validator(s)	**NIAP CCEVS** Margaret Webster-Butler Dr. Jerome Myers

3 Security Policy

The TOE consists of a pair of libraries that can contribute to the enforcement of some security policies. However, the TOE does not by itself implement any security policies. The TOE is intended to be used in an environment that implements security policies that are based upon X.509 certificates and externally provided SPIFs. The TOE performs the processing to determine whether the specified SPIF grants the specified access to the entity identified by the X.509 certificate. The policy that is represented by the SPIF is externally defined. Moreover, the IT Environment is not necessarily obligated to implement the access control recommendation made by the TOE, since there may be other policies involved in determining the actual access controls within the IT Environment.

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 then a success code is returned. Otherwise an error is indicated.

4 Assumptions and Clarification of Scope

4.1 Usage Assumptions

The evaluation made the following assumption concerning product usage:

> 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 the site-specific security requirements.
>
> Applications or programs that use the TOE will correctly interpret the results returned by the ACDF and implement the access control decision accordingly.
>
> The TOE will be integrated in an application and administered by users who are responsible.
>
> SPIFs, X.509 certificates and other inputs into the ACL contain valid security labels and authorizations
>
> The TOE will be installed on a system that provides domain separation and non-bypassibility of the security function provided by the TOE.
>
> The TOE will be installed on a system that is physically protected according to site-specific requirements.

4.2 Clarification of Scope

The TOE is a pair of software libraries that must be integrated into a trusted application to implement any security policies. The TOE itself does not completely implement any security policies; it makes an access control decision recommendation that must be enforced by the IT Environment to actually address the security threats.

The TOE includes software development guidance to ensure that the libraries are properly integrated into an application in a manner that will meet the assumptions listed in the previous section. It is likely that some additional code analysis will have to be performed on the IT Environment to ensure that the correct operation of the TOE is not interfered with by the environment. It is the integrators/certifiers/accreditors responsibility to determine that these conditions are met for the specific integrated application..

The libraries that comprise the TOE present some interfaces to an integrator that are not intended to be used in the evaluated configuration. The only external library interface covered by this evaluation is the ACDF. There are library interfaces present to perform ASN.1 unwrapping of certificates and many of the activities necessary to determine whether a digital certificate is correctly formatted and currently valid. The security functionality potentially provided by these interfaces (except when it is used internally between components of the TOE) is not part of this evaluation.

The TOE was developed using the C++ programming language. Hence, the libraries could be ported to a wide variety of platforms. However, the actual product evaluation was only performed on a Windows 2000 platform using the Visual Studio C++ 6.0 Compiler. Since the tools used for compiling the libraries were included in the evaluation, this compiler must be used for the results of the evaluation to be considered valid.

5 Architectural Information

The TOE is comprised of two software libraries (ACL and eSNACC) 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 Access Control Decision Function 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 authorizations are commensurate with values in a security label, the ACL uses Security Policy Information Files (SPIFs). Figure 1: TOE Boundary illustrates the relationship between the two libraries and applications that interface with those libraries.

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 (now part of BAE Systems). 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 libraries were designed using object oriented techniques. The ACL DLL file is named acl.dll and the eSNACC DLL file is named cpppasn1.dll.
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 can 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. The ACDF 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.

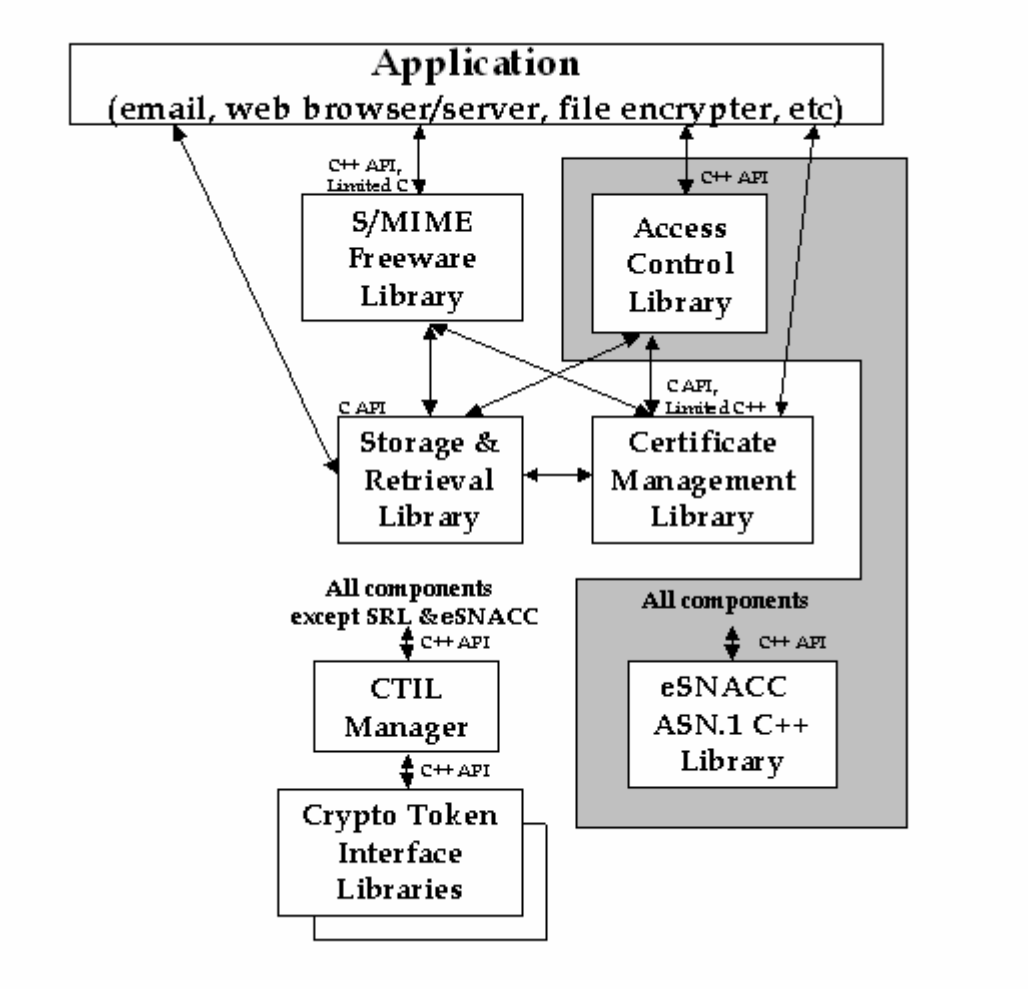

Figure 1: TOE Boundary

6 Delivery and Documentation

The TOE is provided on a CD-R that may be directly obtained from BAE Systems. The CD-R has the identifier ACL Version 2.0.1 printed on the surface of the CD-R. Along with the delivered CD-R, a hard copy of the media thumbprint is provided. The media thumbprint can be used to verify that the software has not been modified. The CD-R contains a single GNU tar and gzip archive containing the software. Also on the CD-R there is an executable “verify.exe” which uses the thumbprint to verify the integrity of the software. The thumbprint hardcopy will indicate the software and revision number that applies to the thumbprint.

The CD_R also contains the following softcopy documentation:

A) Access Control Library Application Programming Interface, Version 2.0, dated 10/31/2001, Document E3-1101-002_acl_api_r2_0;

B) ACL Common Criteria EAL 3 Supplement, Version 2.0.1, dated 9/26/2003, Document ACL_R201_EAL3_Supplement11_0926031;

C) SMP Components Setup Manual Version 2.0.1[11];

7 IT Product Testing

7.1 Developer Testing

The developer maintains a suite of tests for confirming that the product meets its advertised functional requirements. Testing is performed at developer facilities in Annapolis Junction, MD and Herndon, VA. The basic test configuration for the evaluated configuration testing was same as the one illustrated in the architecture diagram in Figure 1: TOE Boundary with the "Application" being a specific application, called Acltool, written by the vendor to serve as a driver for all of the interface tests.

Acltool requires one argument, a configuration filename from which the test cases are read. Acltool provides output to indicate its progress through the test functions it is executing. Each test case is enumerated along with the result of the test case (success or failure). In failure cases a detailed description of the error is indicated. By default if a test case fails, the acltool will report the failure and stop execution. The -e flag can be used to execute all of the test cases, reporting all errors. There is also optional –log flag which re-directs all output from acltool to a log file.

Test documentation including test plans, test procedures, a description of the test configuration, test coverage documentation, expected test results, and actual test results were provided to the CCTL for review. The developers test documentation was provided in the two reports "Access Control Library EAL 3 Software Test Description (ACL STD), June 20, 2003 version 2.0.1" and "Access Control Library EAL3 Software Test Report (ACL STR), June 20, 2003, version 2.0.1". The Test Cases provide a high level description of the functionality tested and test setup. The Test Cases were mapped to one or more Test Procedures. The Test Procedures provided detailed instructions for the tester as well as expected and actual test results.

The evaluators reviewed the developers tests and test results to ensure that the developers testing and test results were appropriate for the evaluated configuration. The evaluators also reviewed the test tool, "acltest", that was used to drive the tests to ensure that the tests were being properly invoked and that the responses were being provided by the TOE. An evaluation team review of all of the security functions and the mapping between security functions and tests confirmed that security functions were appropriately tested by the developer tests.

7.2 Evaluator Testing

Evaluation team testing was conducted on October 13, 2004 at the COACT facility in Columbia, Maryland. The evaluation team performed the following activities during testing:

1. Installation of the TOE
2. Execution of all of the developer's functional tests
3. Independent Testing
4. Vulnerability Testing (AVA_VLA.1)

The evaluation team testing was performed on a similar configuration to that used by the system developers. The testing was performed on a Windows 2000 based PC with the following software:

A) Windows 2000: Operating System

B) ACLTOOL: ACL test tool included on the ACL software CD.

C) ACL Version 2.0.1: Portion of the TOE

D) eSNACC Version 1.3: Portion of the TOE

E) Visual Studio 6.0: C++ compiler

The evaluation team repeated all of the tests in the developer test plan and procedures. The results of the independent testing is documented in the companion document DigitalNet ACL Version 2.0.1 and eSNACC Version 1.3 Independent Evaluator Tests.

A vendor representative was available to facilitate some of the testing. The role of the vendor representative was to facilitate the resolution of any apparent discrepancies between the evaluator's test results and the expected test results. There was only one potential discrepancy noted; an ambiguous message in a return code, that required further discussion with the vendor test team before the evaluators were able to dismiss the issue.

The evaluation team's independent testing independent testing consisted of some variants of the original vendor tests with modified parameters.

Finally, the evaluator performed an analysis of the vendor hypothesized vulnerabilities and associated tests. The vendor developed vulnerability analysis was heavily based upon IT Environment requirements that eliminated most potential vulnerabilities from the scope of the analysis. The evaluator team determined that if the developer guidance was followed for integrating the TOE into an application then the vendor's own vulnerability analysis was thorough and appropriately tested. As a result, there were only a few potential vulnerabilities tested by the evaluators.

The end result of the testing activities was that all tests gave expected (correct) results. The testing found that the product was implemented as described in the functional specification and did not uncover any undocumented interfaces or other security vulnerabilities.

The evaluation team tests and vulnerability tests substantiated the security functional requirements in the ST.

8 Evaluated Configuration

8.1 TOE

This section documents the configuration of the IT product during the evaluation. The TOE consists of the two software libraries: Access Control Library Version 2.1 and eSNACC Version 1.3.

8.1.1 Physical Boundary of TOE

Since the TOE is made up of two software libraries, the physical boundary is the two 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 Figure 1: TOE Boundary. The interfaces to each of the libraries are the Application Programmers Interface (API) calls.

8.1.2 Logical Boundary of TOE

The logical boundary of the TOE is the 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.

The TOE provides the following security service:

Access Control Decision Function on whether a specific form of access is permitted to the owner of a specific certificate based upon a provided SPIF.

8.1.3 Platform for TOE

The TOE was developed using the C++ programming language. Hence, the libraries could be ported to a wide variety of platforms. However, the actual product evaluation was only performed on a Windows 2000 platform using the Visual Studio C++ 6.0 Compiler. Since the tools used for compiling the libraries were included in the evaluation, this platform and compiler must be used for the results of the evaluation to be considered valid.

8.1.4 IT Environment of TOE

Given that this product is a pair of libraries that are intended to be integrated into a trusted application, the primary guidance for administering and installing the product is provided in the software developer guidance, “SMP Components Setup Manual Version 2.0.1”[11]. The evaluation required that the IT product be integrated into an application that had strict requirements on those interfaces that could be used and the manner in

which they could be used. The developer guidance provides the necessary details for the correct integration of the IT product in its evaluated configuration.

The TOE is intended to be used in an IT environment that protects the TOE from modification and restricts the library interfaces that are used. The generic ACL library has other functionality that was explicitly excluded from this evaluation. To meet the requirements for the IT environment, the TOE must be configured so that it is protected by a trusted application that ensures that the TOE is used in accordance with the restrictions placed on the IT environment.

The IT environment must protect the TOE from interference. The correct functioning of the security decision recommendations that are performed by the TOE are contingent on the implementation of the TOE not be overwritten. Since the TOE is a compiled pair of DLLs, this is a requirement that the underlying operating system and trusted portion of the integrated application that uses the DLL to be analyzed and trusted to provide that protection. The IT environment must ensure that any certificates and SPIFs that are passed to the TOE through its evaluated interfaces are valid certificates prior to passing them to the TOE.

The IT environment must ensure that no other interfaces to the TOE are used. This is a requirement on the manner in which a trusted application integrates the TOE to ensure that only the specified interfaces are used by the TOE as well as the trusted application protecting all of the interfaces from being used from outside the trusted application.

Further information on the proper integration of the TOE into an application is available in the publicly available document, “SMP Components Setup Manual Version 2.0.1”[11].

9 Results of the Evaluation

A verdict for an assurance component is determined by the resulting verdicts assigned to the corresponding evaluator action elements. The evaluation was conducted based upon CC, Version 2.1 and CEM, Version 1.0.,

The Evaluation Team assigned a Pass, Fail, or Inconclusive verdict to each work unit of each EAL 3 assurance component. For Fail or Inconclusive work unit verdicts, the Evaluation Team advised the developer of issues requiring resolution or clarification within the evaluation evidence.

In this way, the Evaluation Team assigned an overall Pass verdict to the assurance component only when all of the work units for that component had been assigned a Pass verdict. Section 4, Results of Evaluation, from the document *ACL Version 2.0.1 and eSNACC Version 1.3 Evaluation Technical Report, Document No. F3-0105-2), Dated April 25, 2005 [9]* contain the verdicts of "PASS" for all the work units.

The evaluation determined the product to be Part 2-extended and, as well, meeting the requirements for Part 3, and EAL 3. The details of the evaluation are recorded in the Evaluation Technical Report (ETR), which is controlled by COACT Inc.

10 Validator Comments

The nature of this TOE (two DLLls) does not lend itself well to an isolated evaluation. The product was developed with the intention of integrating it into a family of different applications. Rather than evaluating the TOE multiple times (within each of the specific integrated products,) the evaluation sponsor chose to craft a security target that restricted the evaluation to the two libraries. To accomplish this objective, some very strong assumptions and requirements were placed on the manner in which the libraries would be integrated and used. Ensuring that those assumptions and requirements are met is the responsibility of other parties (integrator, accreditors, etc).

The vulnerability analysis for the TOE was very limited. Almost all potential vulnerabilities were eliminated by placing requirements on the IT environment that ensure that those vulnerabilities could not exist. However, when the product is actually integrated into an application, vulnerability testing on the application will have to be performed to ensure that the IT Environmental constraints are actually met. Hence, the results of the vulnerability analysis for the TOE will not provide much of a reduction in analysis for the integrated application.

The ACL and eSNACC libraries have other interfaces that were not evaluated. The ACL provides several different types of access decision comparisons which were not evaluated and the eSNACC could potentially be used for other ASN.1 encoding/decoding applications besides those used by the ACL. The use of those interfaces to perform other certificate and SPIF related activities, such as validation of the certificates were not included in this evaluation. The validator believes that it is likely that an integrator using the ACL will want to use some of those additional features to ensure that the IT Environment meets the requirements for the ACL. Unfortunately, the use of those interfaces would require further code analysis, testing, and vulnerability analysis that were not included in this evaluation. Fortunately, the library source code

and documentation is publicly available and the implementation is simple enough that it would not be difficult for the responsible parties to acquire access to the necessary information and to perform additional analysis. However, in the process of performing those activities, the integrator or system certifier would end up revisiting much of the analysis already performed for this evaluation.

The freeware library was developed with U.S. Government funds. The developer maintains a public web site where the most recent versions of the library and associated product documentation may be downloaded free of charge. The address of the web site is ",http://www.digitalnet.com/knowledge/acl_lib.htm". This web site includes additional product information, such as test tools, that is not typically available for commercial products. However, since the evaluated version of the product is not necessarily the most recent version of the product, the user of this web site must remember that the information on this web site does not necessarily contain the evaluated version of the product and the web site is not the approved method for obtaining the evaluated product. The CD-R rather than the web site is the only approved means of delivery of the evaluated TOE.

11 Security Target

The Security Target, "ACL Version 2.0.1 and eSNACC Version 1.3 Security Target, Revision 10, dated April 15, 2005" [9] is included here by reference.

12 Glossary

12.1 Definition of Terms

Certificate, Digital
An electronic identification card for a user or device. Digital certificates are distributed, or granted, by certificate authorities, and ensure that the user or device is who/what they claim to be. Digital certificate holders have a public and private key pair, which can be used to sign messages (authenticate the sender), and decrypting incoming messages (ensuring only the certificate holder can decode the encrypted message).

12.2 Definition of Acronyms

AC	Attribute Certificate
ACDF	Access Control Decision Function
ACL	Access Control Library
API	Application Program Interface
ASN.1	Abstract Syntax Notation One
CC	Common Criteria
CCEVS	Common Criteria Evaluation and Validation Scheme
CCTL	Common Evaluation Testing Laboratory
CEM	Common Evaluation Methodology

DLL	Dynamically Linked Library
EAL	Evaluation Assurance Level
eSNACC	Enhanced Sample Neufeld ASN.1 to C/C++ Compiler
ETR	Evaluation Technical Report
HTTP	Hypertext Transport Protocol
IT	Information Technology
NIAP	National Information Assurance Program
NIST	National Institute of Science & Technology
NSA	National Security Agency
NVLAP	National Voluntary Laboratory Assessment Program
PP	Protection Profile
PRBAC	Partition Rule Based Access Control
SDN	Secure Data Network
SMP	Secure Message Protocol
SPIF	Security Policy Information File
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functions
TSP	TOE Security Policy

13 Bibliography

[1] Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model, dated August 1999, Version 2.1.

[2] Common Criteria for Information Technology Security Evaluation – Part 2: Security functional requirements, dated August 1999, Version 2.1.

[3] Common Criteria for Information Technology Security Evaluation – Part 2: Annexes, dated August 1999, Version 2.1.

[4] Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance requirements, dated August 1999, Version 2.1.

[5] Common Evaluation Methodology for Information Technology Security – Part 1: Introduction and general model, dated 1 November 1998, version 0.6

[6] Common Evaluation Methodology for Information Technology Security – Part 2: Evaluation Methodology, dated August 1999, version 1.0.

[7] NIAP Common Criteria Evaluation and Validation Scheme for IT Security, Guidance to Common Criteria Testing Laboratories, Version 1.0, March 20, 2001.

[8] Common Criteria Evaluation and Validation Scheme for Information Technology Security Guidance to Validators of IT Security Evaluations, Scheme Publication #3, Version 1.0, February 2002

[9] ACL Version 2.0.1 and eSNACC Version 1.3 Evaluation Technical Report , Document No. F3-0105-006(2), Dated April 25, 2005

[10] ACL Version 2.0.1 and eSNACC Version 1.3 Security Target, Revision 10, dated April 15, 2005

[11] SMP Components Setup Manual Version 2.0.1.