Sybase
Adaptive Server Enterprise 15.0.1
Security Target

Version 1.0

9/18/2007

Prepared for:

Sybase, Inc.

One Sybase Drive
Dublin, CA 94568

Prepared By:

Science Applications International Corporation

Common Criteria Testing Laboratory

7125 Columbia Gateway Drive, Suite 300
Columbia, MD 21046

LIST OF TABLES

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is Adaptive Server Enterprise 15.0.1 provided by Sybase, Inc.. Adaptive Server Enterprise is a relational database management system (RDBMS) server that operates in the context of a commercial operating system, providing services to local and remote clients via the Tabular Data Stream (TDS) protocol.

The Security Target contains the following additional sections:

- TOE Description (Section 2)
- Security Environment (Section 3)
- Security Objectives (Section 4)
- IT Security Requirements (Section 5)
- TOE Summary Specification (Section 6)
- Protection Profile Claims (Section 7)
- Rationale (Section 8).

1.1 Security Target, TOE and CC Identification

ST Title – Sybase Adaptive Server Enterprise 15.0.1 Security Target

ST Version – Version 1.0

ST Date – 9/18/2007

TOE Identification – Sybase Adaptive Server Enterprise, Version 15.0.1

CC Identification – Common Criteria for Information Technology Security Evaluation, Version 2.3, August 2005

1.2 Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional requirements, Version 2.3, August 2005.
    - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance requirements, Version 2.3, August 2005.
    - Part 3 Conformant
    - EAL 4 augmented with ALC_FLR.2

1.3 Conventions, Terminology, Acronyms

This section specifies the formatting information used in the Security Target.

1.3.1 Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
    - Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a letter in parenthesis placed at the end of the component. For example FDP_ACC.1a and FDP_ACC.1b indicate that the ST includes two iterations of the FDP_ACC.1 requirement, a and b.
    - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]).
    - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
    - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").
- Explicit Security Functional Requirements are identified with the following symbol suffix: "_EXP", for example FTA_MCS_EXP.
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

1.3.2 Acronyms

- ACL – Access Control List
- ASE - Adaptive Server Enterprise
- CM – Configuration Management
- DAC – Discretionary Access Control
- DBMS – Database Management System
- DBO – Database Owner
- DDL – Data Definition Language
- DML – Data Manipulation Language
- IT – Information Technology
- PP – Protection Profile
- RDBMS – Relational Database Management System
- SAR – Security Assurance Requirement
- SFR – Security Functional Requirement
- SQL – Standard Query Language
- SUID – Server-wide User Identifier
- ST – Security Target
- TBO – Table Owner
- TDS – Tabular Data Stream
- TOE – Target of Evaluation
- TSF – TOE Security Functions
- TSP – TOE Security Policy

- UID – Database-specific User Identifier

2. TOE Description

The Target of Evaluation (TOE) is Sybase Adaptive Server Enterprise, Version 15.0.1, configured and operated according to the guidance documents identified later in this Security Target.

Adaptive Server Enterprise (ASE) is a Database Management System (DBMS) designed to execute as a set of applications in the context of commercially available operating systems, specifically Microsoft Windows 2000 (SP4) for x86, Microsoft Windows Server 2003 for x86, Sun Solaris Version 8 for sparc (32- and 64-bit), Sun Solaris Version 9 for sparc (32- and 64-bit), Sun Solaris Version 10 for sparc (32- and 64-bit), IBM AIX 5L Version 5.2 (64-bit), Hewlett-Packard HP-UX 11i v1 for PA-risc (64-bit), Hewlett-Packard HP-UX 11i v2 for PA-risc (64-bit), Red Hat Enterprise Linux 3.0 for x86and Red Hat Enterprise Linux 4.0 for x86.

Note that ASE version 15.0.1 is a revised version of the previously evaluated ASE Version 12.5.2. Among a number of non-security relevant feature additions and modifications (such as partitioned databases on a given server and a new query processing engine), ASE version 15.0.1 includes resource governor enhancements and the ability to encrypt database columns.

2.1 TOE Overview

The ASE Server runs as an application on top of an operating system and depends on the services exported by the operating system to function. ASE uses operating system services for process creation and manipulation; device and file processing; shared memory creation and manipulation; and security requests such as inter-process communication. The hardware upon which the operating system runs is completely transparent to ASE - ASE sees only the operating system's user interfaces.

The ASE Server is one or more operating system processes that service client requests. Multiple processes can be configured to enhance performance on multiprocessor systems. An ASE process has two distinct components, a DBMS component and a kernel component. The DBMS component manages the processing of SQL statements (data manipulation language - DML, data definition language - DDL, stored procedures and administrative commands), accesses data in a database, and manages different types of Server resources. The kernel component performs low-level functions for the DBMS component, such as task and engine management; network and disk I/O; and low-level memory management. Note that the TDS engine, that part of ASE that processes a TDS request, also uses the kernel component for low-level services.

All of the ASE processes attach to one or more shared memory segments. The shared memory contains data structures that relate to task management and operating system services, caches of database buffers, object descriptors, and other resources (e.g., other caches, queues, and stream I/O buffers) required to manage and process database commands.

Each ASE process manages multiple ASE tasks. A task is a thread of execution within the Server. Each client is associated with its own ASE task. In addition, there are several system tasks that perform specific services (e.g., tasks to write buffers to disk, tasks to write audit data to disk, and tasks to communicate with the network.).

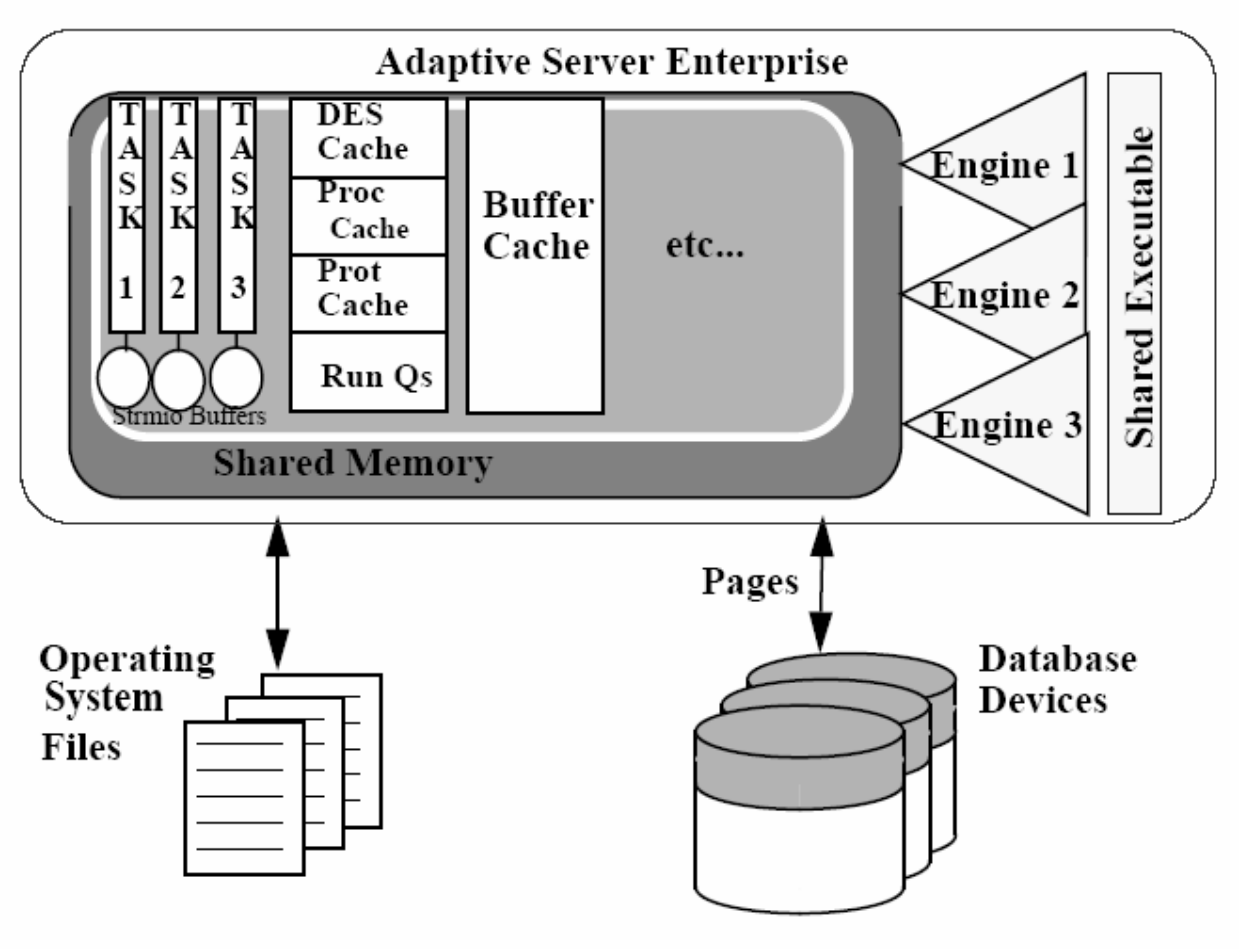

Figure 1 Target of Evaluation

2.2 TOE Architecture

The components that make up the majority of the Target of Evaluation (TOE) are depicted in Figure 1. The components are:

- Engine - an operating system process that is running the ASE executable. Logically there is a single server that may consist of multiple engines on a multi-processor machine.
- Shared memory - used by the engine(s) to manage shared resources such as tasks, data buffers, etc.
- Operating system files - A number of operating system files are used by the Server for configuration.
- Devices - operating system files or disk partitions used to store the databases (metadata and data) accessed and managed by the Server.

In addition to the components identified above, the TOE includes an additional program, *isql*, as well as several additional binaries that provide the interfaces necessary for TOE administration. For the evaluated configuration of the TOE, only those binaries that affect the security functions will be further identified herein.

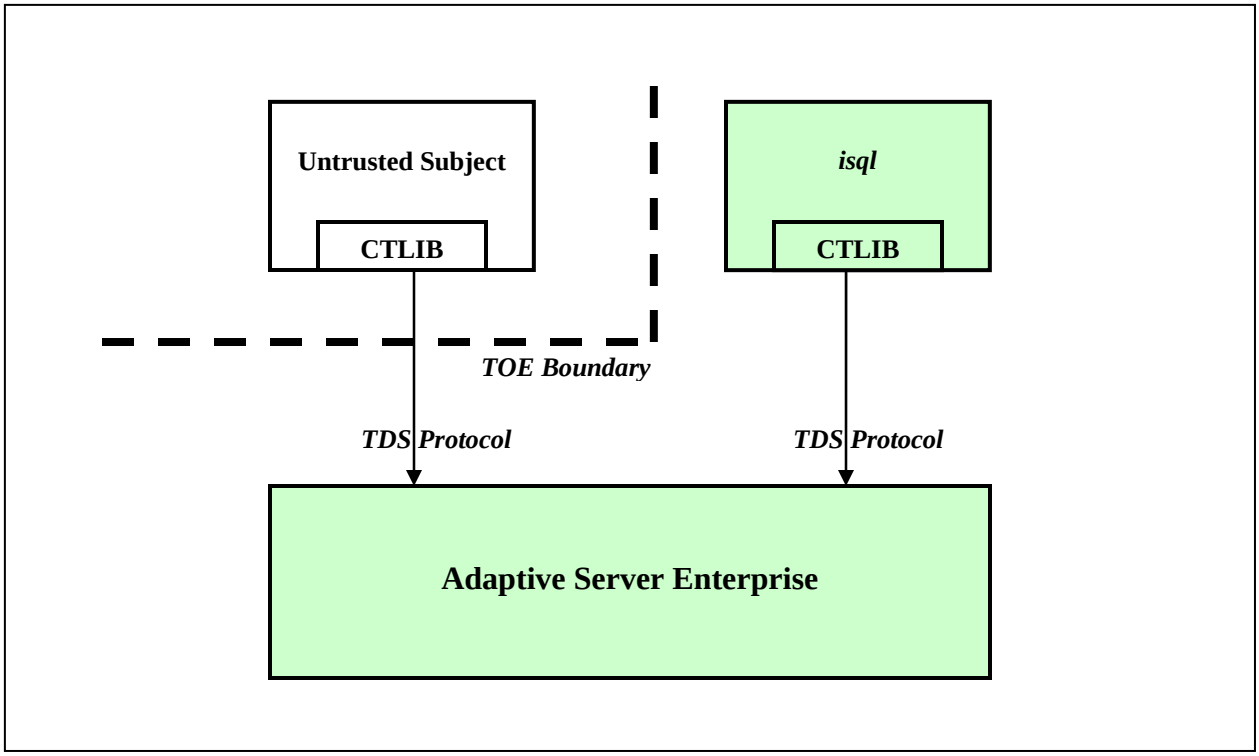

Figure 2 ASE Interfaces

2.2.1 Physical Boundaries

The only mechanism available to communicate with the ASE Server is the Tabular Data Stream (TDS) Protocol. This interface is shown in Figure 2.

The TDS protocol is used to request Server services. TDS is used by untrusted client processes, via routines in CT Library (CTLIB) and jConnect, to communicate with the Server. Administrators interface with the ASE Server via the *isql* utility program. *isql* uses library routines in CTLIB to interface to TDS, which in turn communicates with the Server over TCP/IP. Note that *isql* and its supporting libraries is also part of the TOE inasmuch as it provides required administration functions.

TDS communicates with the Server via messages. The functionality of the Server is reflected by the type of TDS messages and their contents. TDS is discussed in more detail in the TDS Specification.

2.2.2 Logical Boundaries

The TOE logically supports the following security functions at its interfaces:

- Security audit
- Cryptographic support,
- User data protection
- Identification and authentication
- Security Management,
- Protection of the TSF,
- Resource utilization, and
- TOE access

2.2.2.1 Security audit

ASE has an audit mechanism that is invoked for access checks, authentication attempts, administrator functions, and at other times during its operation. When invoked, the date, time, responsible individual and other details describing the event are recorded to the audit trail.

The Audit log is stored as tables within ASE itself so that audit records can be protected from unauthorized access or modification. Furthermore, the SQL select command provided by ASE can be used by System Security Officers to effectively review the audit trail, including searching and sorting by user identities and other audit record attributes.

2.2.2.2 Cryptographic support

ASE supports the ability to encrypt data at the column level. Encryption of only the sensitive data minimizes processing overhead as compared to encrypting an entire database. In order to accomplish this, ASE includes a FIPS 140-2 validated cryptographic module, Certicom Security Builder GSE, to perform cryptographic operations.

SQL statements are available to create applicable encryption keys and specify columns for encryption. ASE handles key generation and storage and also provides System Security Officers the ability destroy keys that are no longer needed. Encryption and decryption of data occurs automatically and transparently as data is written to and read from encrypted columns. No client application changes are required.

2.2.2.3 User data protection

ASE implements a Discretionary Access Control Policy over applicable database objects - databases, tables, views, and stored procedures. Note that there are other database objects that are either always private, always public, or are part of one of the afore-mentioned objects. In each case, the objects each have an owner which is initially the creator of the object. Object owners have special permissions, while other users can subsequently be granted specific access permissions based on user identity, group memberships and active roles allowing applicable operations on objects.

ASE also implements a Policy-based Access Control Policy over the content of database tables. This policy controls access based on Application Contexts of the current subject in conjunction with Access Rules associated with columns in database tables. This policy effectively allows access to be controlled on very specific and widely varying information about users.

2.2.2.4 Identification and authentication

ASE provides its own identification and authentication mechanism in addition to the underlying operating system. Users must provide a valid username and password before they can access any security-related functions. Once identified and authenticated, all subsequent actions are associated with that user and policy decisions are based on the users identity, group memberships and active roles.

2.2.2.5 Security management

ASE provides functions necessary to manage users and associated privileges, access permissions, and other security functions such as audit. The functions are restricted based on Discretionary Access Control Policy rules including role restrictions. While all of the administrative functions are available through and restricted at the TDS ASE Server interface, an application (*isql*) are provided to support ASE administrators.

ASE defines a number of system-defined roles - System Administrator (SA), System Security Officer (SSO), Operator, etc.. Otherwise, there are users of the TOE of which the Database Owner (DBO) has special rights with regard to their own database. However, of these roles, only the SA and SSO have any special rights with respect to the security functions claimed in this Security Target. While it seems the DBO has special rights, their rights are all based on access permissions associated with the database they own.

2.2.2.6 Protection of the TSF

ASE protects itself and ensures that its policies are enforced in a number of ways. While there is dependence on the underlying operating system to separate its process constructs, enforce file and memory access restrictions, and to provide communication services, ASE protects itself by keeping its context separate from that of its users and also

by making effective use of the operating system mechanisms to ensure that memory and files used by ASE have the appropriate access settings. Furthermore, ASE interacts with users through well-defined interfaces designed to ensure that the ASE security policies are always enforced.

2.2.2.7 Resource utilization

ASE provides resource limits to help System Administrators prevent queries and transactions from monopolizing server resources. Specifically, System Administrators can configure ASE to prevent queries and transactions that: exceed estimated or actual I/O costs, return too many rows, exceed the temporary database space allocated, and/or exceed a specified elapsed time.

When a System Administrator configures a resource limit, all current users are immediately subject to the new limits unless resource limits are not enabled. If resource limits are not enabled, the System Administrator is notified that the configured limits would become effective when resource limits are enabled for the server.

2.2.2.8 TOE access

ASE allows System Security Officers to construct login triggers that can be used to restrict logins to a specific number of sessions and specified times. ASE also allows System Security Officers to restrict access based on user identities.

2.3 TOE Documentation

Sybase offers a series of documents that describe the installation process for Adaptive Server Enterprise as well as guidance for subsequent use and administration of the applicable security features. Refer to Section 6 for information about these and other documents associated with Adaptive Server Enterprise.

3. Security Environment

The security environment for the functions addressed by this specification includes threats, security policies, and usage assumptions, as discussed below.

3.1 Organizational Policies

P.ACCOUNTABILITY	The users of the TOE shall be held accountable for their actions within the TOE.
P.AUTHORIZATION	The TOE shall limit the extent of each user's abilities in accordance with the TSP.
P.AUTHORIZED_USERS	Access controls will ensure that only those users who have been authorized to access the protected information within the TOE will be able to do so.
P.I_AND_A	All users must be identified and authenticated prior to accessing any controlled resources with the exception of public objects.
P.NEED_TO_KNOW	The TOE must limit the access to information in protected resources to those authorized users who have a need to know that information.
P.ROLES	The TOE shall provide a system administrator and a system security officer role for secure administration of the TOE. This role shall be separate and distinct from other authorized users.

3.2 Threats

T.ADMIN_ERROR	A system administrator or a system security officer may incorrectly install or configure the TOE resulting in ineffective security mechanisms.
T.AUDIT_COMPROMISE	A process or user may cause audit data to be inappropriately accessed (viewed, modified or deleted), or prevent future records from being recorded, thus masking an attacker's actions.
T.MASQUERADE	An unauthorized user, process, or external IT entity may masquerade as an authorized entity to gain access to data or TOE resources.
T.RAW_ACCESS	An unauthorized user may gain access to sensitive information after obtaining access to the raw database storage medium.[1]
T.RESIDUAL_DATA	A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.

[1] Note that this threat is fully addressed only under specific circumstances. First, the encryption feature of the product must be enabled. Second, the keys used to protect the data must be kept physically separate from the data that is protected. These aspects of the TOE are configurable by the TOE administrators.

T.RESOURCE	An authenticated database user might consume excessive global database resources in a way which compromises the ability of other database users to access the DBMS.
T.SYSACC	A malicious process or user may gain unauthorized access to the a system administrator or a system security officer account, or that of other trusted personnel.
T.TSF_COMPROMISE	A malicious user or process may cause TOE data to be inappropriately accessed (viewed, modified or deleted).
T.UNAUTH_ACCESS	A user may gain unauthorized access (view, modify, delete) to user data.
T.UNDETECTED_ACTIONS	Failure of the IT operating system to detect and record attempts to perform unauthorized actions may occur.
T.UNIDENTIFIED_ACTIONS	Failure of the system administrator or system security officer to identify and act upon attempts to perform unauthorized actions may occur.

3.3 Assumptions

A.NO_EVIL	System administrators and system security officers are non-hostile, appropriately trained and follow all administrator guidance.
A.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on DBMS servers, other than those services necessary for the operation, administration and support of the DBMS.
A.PHYSICAL	It is assumed that appropriate physical security is provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.
A.ROBUST_ENVIRONMENT	It is assumed that the IT environment provides support commensurate with the expectations of the TOE.
A.NETWORK	It is assumed that the environment protects network communication media appropriately.

4. Security Objectives

This section defines the security objectives for the TOE and its environment. These objectives are suitable to counter all identified threats and cover all identified organizational security policies and assumptions. The TOE security objectives are identified with 'O.' inserted at the beginning of the name and the environment objectives are identified with 'OE.' inserted at the beginning of the name.

4.1 Security Objectives for the TOE

O.ACCESS	The TOE will ensure that users gain only authorized access to it and to the resources that it controls.
O.ADMIN_ROLE	The TOE will provide system administrator and system security officer roles to isolate administrative actions.
O.AUDIT_GENERATION	The TOE will provide the capability to detect and create records of security relevant events associated with users.
O.AUDIT_PROTECTION	The TOE will provide the capability to protect audit information.
O.AUDIT_REVIEW	The TOE will provide the capability to selectively view audit information.
O.CRYPTO	The TOE will provide the capability to encrypt sensitive data..
O.DISCRETIONARY_ACCESS	The TOE will control access to resources based upon the identity of users, roles active in a user session, group membership of users, object ownership and access control lists.
O.INTERNAL_TOE_DOMAINS	The TSF will maintain internal domains for separation of data and queries belonging to concurrent users.
O.MANAGE	The TOE will provide all the functions and facilities necessary to support the system administrators and system security officers in their management of the security of the TOE.
O.PROTECT	The TOE will provide mechanisms to protect user data and resources.
O.RESIDUAL_INFORMATION	The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.
O.RESOURCE	The TOE must provide the means of controlling the consumption of database resources by authorized users of the TOE.
O.TOE_PROTECTION	The TOE will protect itself and its assets from external interference or tampering.
O.USER_AUTHENTICATION	The TOE will verify the claimed identity of users.

O.USER_IDENTIFICATION	The TOE will uniquely identify users.

4.2 Security Objectives for the IT Environment

OE.TIME	The IT environment will provide a time source that provides reliable time stamps.
OE.TOE_PROTECTION	The IT environment will provide protection to the TOE and its assets from external interference, tampering, and disclosure.

4.3 Security Objectives for the Environment

OE.CONFIG	The TOE will be installed, configured, managed and maintained in accordance with its guidance documentation and applicable security policies and procedures by appropriately trained and trusted administrator personnel.
OE.NO_GENERAL_PURPOSE	There will be no general-purpose computing capabilities (e.g., compilers or user applications) available on DBMS servers, other than those services necessary for the operation, administration and support of the DBMS.
OE.PHYSICAL	Physical security will be provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.
OE.TRUST_IT	Each IT entity the TOE relies on for security functions will be installed, configured, managed, maintained and provide the applicable security functions in a manner appropriate to the IT entity, and consistent with the security policy of the TOE and the relationship between them.
OE.NETWORK	The environment must protect network traffic to and from the TOE from unauthorized disclosure.

5. IT Security Requirements

This section defines the security functional and security assurance requirements for the TOE and associated IT environment components. Note that in addition to these requirements, Adaptive Server Enterprise also satisfies a minimum strength of function 'SOF-medium'. The only applicable (i.e., probabilistic or permutational) security functions are FIA_SOS.1, FIA_UAU.2, and FIA_UID.2 which are all levied on the TOE.

5.1 TOE Security Functional Requirements

The following table describes the SFRs that are candidates to be satisfied by Adaptive Server Enterprise.

Requirement Class	Requirement Component
FAU: Security audit	FAU_GEN.1: Audit data generation
	FAU_GEN.2: User identity association
	FAU_SAR.1: Audit review
	FAU_SAR.2: Restricted audit review
	FAU_SAR.3: Selectable audit review
	FAU_SEL.1: Selective audit
	FAU_STG.1: Protected audit trail storage
	FAU_STG.3: Action in case of possible audit data loss
FCS: Cryptographic support	FCS_CKM.1: Cryptographic key generation
	FCS_CKM.4: Cryptographic key destruction
	FCS_COP.1a: Cryptographic operation
	FCS_COP.1b: Cryptographic operation
FDP: User data protection	FDP_ACC.1a: Subset access control
	FDP_ACC.1b: Subset access control
	FDP_ACF.1a: Security attribute based access control
	FDP_ACF.1b: Security attribute based access control
	FDP_RIP.2a: Full residual information protection
FIA: Identification and authentication	FIA_AFL.1: Authentication failure handling
	FIA_ATD.1: User attribute definition
	FIA_SOS.1: Verification of secrets
	FIA_UAU.2: User authentication before any action
	FIA_UAU.7: Protected authentication feedback
	FIA_UID.2: User identification before any action
	FIA_USB.1: User-subject binding
FMT: Security management	FMT_MOF.1: Management of security functions behaviour
	FMT_MSA.1: Management of security attributes
	FMT_MSA.2: Secure security attributes
	FMT_MSA.3: Static attribute initialization
	FMT_MTD.1a: Management of TSF data
	FMT_MTD.1b: Management of TSF data
	FMT_MTD.1c: Management of TSF data
	FMT_MTD.1d: Management of TSF data
	FMT_REV.1a: Revocation
	FMT_REV.1b: Revocation
	FMT_SMF.1: Specification of Management Functions
	FMT_SMR.1: Security roles
FPT: Protection of the TSF	FPT_RVM.1a: Non-bypassability of the TSP
	FPT_SEP.1a: TSF domain separation
FRU: Resource utilization	FRU_RSA.1: Maximum quotas

Requirement Class	Requirement Component
FTA: TOE access	FTA_MCS_EXP.1: Basic limitation on multiple concurrent sessions
	FTA_TSE.1: TOE session establishment

Table 1 TOE Security Functional Components

5.1.1 Security audit (FAU)

5.1.1.1 Audit data generation (FAU_GEN.1)

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events: a) Start-up and shutdown of the audit functions; b) All auditable events for the ***[not specified]*** level of audit; and c) **[the auditable actions identified in the following table]**.

Requirement Component	Auditable Action
FAU_GEN.1: Audit data generation	None
FAU_GEN.2: User identity association	None
FAU_SAR.1: Audit review	None
FAU_SAR.2: Restricted audit review	None
FAU_SAR.3: Selectable audit review	None
FAU_SEL.1: Selective audit	All modifications to the audit configuration that occur while the audit collection functions are operating.
FAU_STG.1: Protected audit trail storage	None
FAU_STG.3: Action in case of possible audit data loss	None
FCS_CKM.1: Cryptographic key generation	All key creation and modification commands.
FCS_CKM.4: Cryptographic key destruction	All key destruction commands.
FCS_COP.1a: Cryptographic operation	None
FCS_COP.1b: Cryptographic operation	None
FDP_ACC.1a: Subset access control	None
FDP_ACC.1b: Subset access control	None
FDP_ACF.1a: Security attribute based access control	Attempts to perform an operation on an object covered by the SFP.
FDP_ACF.1b: Security attribute based access control	Attempts to perform an operation on an object covered by the SFP.
FDP_RIP.2a: Full residual information protection	None
FIA_AFL.1: Authentication failure handling	The reaching of the threshold for the unsuccessful authentication attempts and the actions (e.g. disabling of a terminal) taken and the subsequent, if appropriate, restoration to the normal state (e.g. re-enabling of a terminal).
FIA_ATD.1: User attribute definition	None
FIA_SOS.1: Verification of secrets	Rejection by the TSF of any tested secret.
FIA_UAU.2: User authentication before any action	Unsuccessful use of the authentication mechanism.
FIA_UID.2: User identification before any action	Unsuccessful use of the user identification mechanism, including the user identity provided.
FIA_UAU.7: Protected authentication feedback	None
FIA_USB.1: User-subject binding	Unsuccessful binding of user security attributes to a subject (e.g. creation of a subject).
FMT_MOF.1: Management of security	None

Requirement Component	Auditable Action
functions behaviour	
FMT_MSA.1: Management of security attributes	None
FMT_MSA.2: Secure security attributes	All offered and rejected values for a security attribute.
FMT_MSA.3: Static attribute initialization	None
FMT_MTD.1a: Management of TSF data	None
FMT_MTD.1b: Management of TSF data	None
FMT_MTD.1c: Management of TSF data	None
FMT_MTD.1d: Management of TSF data	None
FMT_REV.1a: Revocation	None
FMT_REV.1b: Revocation	None
FMT_SMF.1: Specification of Management Functions	Use of the management functions.
FMT_SMR.1: Security roles	Modifications to the group of users that are part of a role.
FPT_RVM.1a: Non-bypassability of the TSP	None
FPT_SEP.1a: TSF domain separation	None
FRU_RSA.1: Maximum quotas	None
FTA_MCS_EXP.1: Basic limitation on multiple concurrent sessions	None
FTA_TSE.1: TOE session establishment	None

Table 2 TOE Security Audit Events

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information: a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, **[no additional information]**

5.1.1.2 User identity association (FAU_GEN.2)

FAU_GEN.2.1 The TSF shall be able to associate each auditable event with the identity of the user that caused the event.

5.1.1.3 Audit review (FAU_SAR.1)

FAU_SAR.1.1 The TSF shall provide **[the System Security Officer]** with the capability to read **[all audit information]** from the audit records.

FAU_SAR.1.2 The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

5.1.1.4 Restricted audit review (FAU_SAR.2)

FAU_SAR.2.1 The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.

5.1.1.5 Selectable audit review (FAU_SAR.3)

FAU_SAR.3.1 The TSF shall provide the ability to perform **[*searches* and *sorting*]** of audit data based on **[any audit record contents]**.

5.1.1.6 Selective audit (FAU_SEL.1)

FAU_SEL.1.1 The TSF shall be able to include or exclude auditable events from the set of audited events based on the following attributes: a) **[*event type* and]** b) **[access attempts by individual users]**.

5.1.1.7 Protected audit trail storage (FAU_STG.1)

FAU_STG.1.1 The TSF shall protect the stored audit records from unauthorized deletion.

FAU_STG.1.2 The TSF shall be able to **[*prevent*]** unauthorized modifications to the audit records in the audit trail.

5.1.1.8 Action in case of possible audit data loss (FAU_STG.3)

FAU_STG.3.1 The TSF shall take **[action to either prevent additional auditable events or discard old audit records in order to create new records, at the discretion of the System Security Officer]** if the audit trail exceeds **[its maximum capacity]**.

5.1.2 Cryptographic support (FCS)

5.1.2.1 Cryptographic key generation (FCS_CKM.1)

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**RNG (random key generation), KDF (key derivation function)**] and specified cryptographic key sizes [**128, 192, or 256 bits**] that meet the following: [**ANSI X9.62, IEEE 1363-2000 KDF1 based on SHA-1**].

5.1.2.2 Cryptographic key destruction (FCS_CKM.4)

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [**zero out the memory locations containing raw key values immediately after use and delete encryption keys from disk when directed by a System Security Officer**] that meets the following: [**no standard**].

Application note: This security functional requirement is intended to accurately convey what the TOE does relative to key destruction; specifically it writes zeroes over keys stored in memory and simply deletes files that contain keys when no longer needed.

5.1.2.3 Cryptographic operation (FCS_COP.1a)

FCS_COP.1a.1 The TSF shall perform [**encryption and decryption of data in applicable database columns and of column encryption keys**] in accordance with a specified cryptographic algorithm [**Advanced Encryption Standard (AES)**] and cryptographic key sizes [**128, 192, or 256 bits**] that meet the following: [**FIPS 197**].

5.1.2.4 Cryptographic operation (FCS_COP.1b)

FCS_COP.1b.1 The TSF shall perform [**hashing in support of KDF key generation**] in accordance with a specified cryptographic algorithm [**SHA-1**] and cryptographic key sizes [**not applicable**] that meet the following: [**FIPS 180-2**].

5.1.3 User data protection (FDP)

5.1.3.1 Subset access control (FDP_ACC.1a)

FDP_ACC.1a.1 The TSF shall enforce the **[Discretionary Access Control Policy]** on **[all database subjects; the following database objects: databases, tables, views, stored procedures, and encryption keys; and, all operations on the identified database objects by database subjects]**.

5.1.3.2 Subset access control (FDP_ACC.1b)

FDP_ACC.1b.1 The TSF shall enforce the **[Policy-based Access Control Policy]** on **[all database subjects; the following database objects: rows; and, the following database operations on rows by database subjects: select, update, and delete]**.

5.1.3.3 Security attribute based access control (FDP_ACF.1a)

FDP_ACF.1a.1 The TSF shall enforce the **[Discretionary Access Control Policy]** to objects based on the following: **[database subject attributes: user identity, group membership and active roles; database object attributes: object owner; and access control lists (ACLs)]**.

FDP_ACF.1a.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **[**

a) if the user identity is equal to the System Administrator, the requested access is allowed; or

b) if the ACL grants the requesting user identity the requested access, the requested access is allowed; or

c) if the user is a member of a group which has been granted access and the ACL does not explicitly revoke the requested access from the requesting user identity, the requested access is allowed; or

d) if the group public has been granted the requested access, and the ACL explicitly does not revoke the requested access from the user's group or the user, the requested access is allowed; or

e) if the user identity has a role active and the ACL grants the role (or a role contained in that role) the requested access, the requested access is allowed; or

f) otherwise access is denied, unless access is explicitly authorized in accordance with the rules specified in FDP_ACF.1a.3.].

FDP_ACF.1a.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **[**

a) if the database subject is a System Administrator, the requested access is allowed; or

b) if an object is accessed via a stored procedure and the owner of the stored procedure owns the object].

FDP_ACF.1a.4 The TSF shall explicitly deny access of subjects to objects based on the **[there are no explicit access denial rules]**.

5.1.3.4 Security attribute based access control (FDP_ACF.1b)

FDP_ACF.1b.1 The TSF shall enforce the **[Policy-based Access Control Policy]** to objects based on the following: **[database subject attributes: Application Context; and, database object attributes: Access Rules]**.

FDP_ACF.1b.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **[**

a) if there is no Access Rule associated with a table column access is allowed; or

b) if the Access Rule allows access to the specified row(s) based on the Application Context access is allowed; or

c) otherwise access is denied.].

FDP_ACF.1b.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **[there are no explicit access authorization rules]**.

FDP_ACF.1b.4 The TSF shall explicitly deny access of subjects to objects based on the **[there are no explicit access denial rules]**.

5.1.3.5 Full residual information protection (FDP_RIP.2a)

FDP_RIP.2a.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the **[*allocation of the resource to*]** all objects.

5.1.4 Identification and authentication (FIA)

5.1.4.1 Authentication failure handling (FIA_AFL.1)

FIA_AFL.1.1 The TSF shall detect when **[*an administrator configurable positive integer within [0 – 32767*[2]*]*]** unsuccessful authentication attempts occur related to **[user identification]**.

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall **[prevent subsequent authentication of the identified user, until re-enabled by a System Security Officer**].

5.1.4.2 User attribute definition (FIA_ATD.1)

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: **[user identity, authentication data, group membership, and active roles]**.

5.1.4.3 Verification of secrets (FIA_SOS.1)

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet **[the following: a) for each attempt to use the authentication mechanisms, the probability that a random attempt will succeed is less than one in 500,000,000,000; and b) any feedback given during each attempt to use the authentication mechanism will reduce the probability of the above metric by only one.]**.

5.1.4.4 User authentication before any action (FIA_UAU.2)

FIA_UAU.2.1 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

5.1.4.5 Protected authentication feedback (FIA_UAU.7)

FIA_UAU.7.1 The TSF shall provide only [**a failure indication that does not identify whether the failure was identity or password related and un-echoed passwords**] to the user while the authentication is in progress.

5.1.4.6 User identification before any action (FIA_UID.2)

FIA_UID.2.1 The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

5.1.4.7 User-subject binding (FIA_USB.1)

FIA_USB.1.1 The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: **[user identity, group membership and active roles]**.

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: **[subject security attributes are derived from TSF data maintained for each defined user after a successful login with the defined user identity]**.

FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: **[a) a System Security Officer can change their identity; b) a user can activate and deactivate roles assigned to them; and c) a System Security Officer can explicitly grant a user the ability to change their identity within limits established by the System Security Officer]**.

[2] In the context of this requirement, a value of '0' indicates no limit to the number of detected failed unsuccessfully authentication attempts.

5.1.5 Security management (FMT)

5.1.5.1 Management of security functions behaviour (FMT_MOF.1)

FMT_MOF.1.1 The TSF shall restrict the ability to **[*disable* and *enable*]** the functions **[related to the specification of events to be audited and the cryptographic function]** to **[System Security Officers]**.

5.1.5.2 Management of security attributes (FMT_MSA.1)

FMT_MSA.1.1 The TSF shall enforce the **[Discretionary Access Control Policy]** to restrict the ability to **[*set* and *reset*]** the security attributes **[of database subjects]** to **[System Administrators, System Security Officers, and Database Owners (who can set group assignments that affect their own database only)]**.

5.1.5.3 Secure security attributes (FMT_MSA.2)

FMT_MSA.2.1 The TSF shall ensure that only secure values are accepted for security attributes.

5.1.5.4 Static attribute initialization (FMT_MSA.3)

FMT_MSA.3.1 The TSF shall enforce the **[Discretionary Access Control Policy]** to provide **[*restrictive]*** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the **[no role]** to specify alternative initial values to override the default values when an object or information is created.

5.1.5.5 Management of TSF data (FMT_MTD.1a)

FMT_MTD.1a.1 The TSF shall restrict the ability to **[*include* or *exclude*]** the **[audited events]** to **[System Security Officers]**.

5.1.5.6 Management of TSF data (FMT_MTD.1b)

FMT_MTD.1b.1 The TSF shall restrict the ability to **[*query* and *clear*]** the **[audit records]** to **[System Security Officers]**.

5.1.5.7 Management of TSF data (FMT_MTD.1c)

FMT_MTD.1c.1 The TSF shall restrict the ability to **[*set* and *reset*]** the **[user authentication data]** to **[System Security Officers and users (who can change only their own user authentication data)]**.

5.1.5.8 Management of TSF data (FMT_MTD.1d)

FMT_MTD.1d.1 The TSF shall restrict the ability to **[*set* and *reset*]** the **[encryption password]** to **[System Security Officers]**.

5.1.5.9 Revocation (FMT_REV.1a)

FMT_REV.1a.1 The TSF shall restrict the ability to revoke security attributes associated with the **[*subjects]*** within the TSC to **[System Administrators, System Security Officers, and Database Owners (who can set group assignments that affect their own database only)]**.

FMT_REV.1a.2 The TSF shall enforce the rules **[: the enforcement of subject attribute changes shall take effect the next time the associated user is identified and authenticated by the TSF]**.

5.1.5.10 Revocation (FMT_REV.1b)

FMT_REV.1b.1 The TSF shall restrict the ability to revoke security attributes associated with the **[*objects]*** within the TSC to **[System Administrators and users (only for database objects they own or database objects for which they have been granted subject access permissions allowing them to revoke security attributes)]**.

FMT_REV.1b.2 The TSF shall enforce the rules **[: the enforcement of object attribute changes shall take effect before the next access attempt related to that object]**.

5.1.5.11 Specification of Management Functions (FMT_SMF.1)

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions: **[starting and stopping the audit function, selection of the audited events, review of audit data, enable and disable the cryptographic function, manage the encryption password, creation and destruction of column encryption keys, and management of database subjects and authentication data]**.

5.1.5.12 Security roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles **[System Administrator, System Security Officer, Database Owner and User]**.

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

5.1.6 Protection of the TSF (FPT)

5.1.6.1 Non-bypassability of the TSP (FPT_RVM.1a)

FPT_RVM.1a.1 The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

5.1.6.2 TSF domain separation (FPT_SEP.1a)

FPT_SEP.1a.1 The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.1a.2 The TSF shall enforce separation between the security domains of subjects in the TSC.

5.1.7 Resource utilization (FRU)

5.1.7.1 Maximum quotas (FRU_RSA.1)

FRU_RSA.1.1 The TSF shall enforce maximum quotas of the following resources: [**database query and transaction I/O costs, row accesses, temporary database space, and duration**] that [***subjects***] can use [***over a specified period of time***].

5.1.8 TOE access (FTA)

5.1.8.1 Basic limitation on multiple concurrent sessions (FTA_MCS_EXP.1)

FTA_MCS_EXP.1.1 The TSF shall be able to restrict the maximum number of concurrent sessions that belong to the same user.

FTA_MCS_EXP.1.2 The TSF shall enforce, by default, no limit to the number of sessions per user.

5.1.8.2 TOE session establishment (FTA_TSE.1)

FTA_TSE.1.1 The TSF shall be able to deny session establishment based on **[user identity and time]**.

5.2 IT Environment Security Functional Requirements

The following table describes the SFRs that are candidates to be satisfied by the IT environment of Adaptive Server Enterprise.

Requirement Class	Requirement Component

Requirement Class	Requirement Component
FDP: User data protection	FDP_ACC.1c: Subset access control
	FDP_ACF.1c: Security attribute based access control
	FDP_RIP.2b: Full residual information protection
FPT: Protection of the TSF	FPT_RVM.1b: Non-bypassability of the TSP
	FPT_SEP.1b: TSF domain separation
	FPT_STM.1: Reliable time stamps

Table 3 IT Environment Security Functional Components

5.2.1 User data protection (FDP)

5.2.1.1 Subset access control (FDP_ACC.1c)

FDP_ACC.1c.1 The TSF shall enforce the **[Environment Access Control Policy]** on **[processes; files and shared memory; and, process operations to access files and shared memory].**

5.2.1.2 Security attribute based access control (FDP_ACF.1c)

FDP_ACF.1c.1 The TSF shall enforce the **[Environment Access Control Policy]** to objects based on the following: **[process attributes: process identification attributes; file and shared memory: ownership and access permissions]**.

FDP_ACF.1c.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **[in order to successfully access (for read or write) a file or shared memory, the applicable process must be permitted the requested access (based on its identification and the access permissions associated with the file or shared memory)]**.

FDP_ACF.1c.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **[if the process (based on its identification) is the owner of the file or shared memory, the access operation will always succeed]**.

FDP_ACF.1c.4 The TSF shall explicitly deny access of subjects to objects based on the **[there are no explicit access denial rules]**.

5.2.1.3 Full residual information protection (FDP_RIP.2b)

FDP_RIP.2b.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the **[*allocation of the resource to*]** all objects.

5.2.2 Protection of the TSF (FPT)

5.2.2.1 Non-bypassability of the TSP (FPT_RVM.1b)

FPT_RVM.1b.1 The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

5.2.2.2 TSF domain separation (FPT_SEP.1b)

FPT_SEP.1b.1 The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.1b.2 The TSF shall enforce separation between the security domains of subjects in the TSC.

5.2.2.3 Reliable time stamps (FPT_STM.1)

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps for its own use.

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the EAL 4 augmented with ALC_FLR.2 components as specified in Part 3 of the Common Criteria. No operations are applied to the assurance components.

Requirement Class	Requirement Component
ACM: Configuration management	ACM_AUT.1: Partial CM automation
	ACM_CAP.4: Generation support and acceptance procedures
	ACM_SCP.2: Problem tracking CM coverage
ADO: Delivery and operation	ADO_DEL.2: Detection of modification
	ADO_IGS.1: Installation, generation, and start-up procedures
ADV: Development	ADV_FSP.2: Fully defined external interfaces
	ADV_HLD.2: Security enforcing high-level design
	ADV_IMP.1: Subset of the implementation of the TSF
	ADV_LLD.1: Descriptive low-level design
	ADV_RCR.1: Informal correspondence demonstration
	ADV_SPM.1: Informal TOE security policy model
AGD: Guidance documents	AGD_ADM.1: Administrator guidance
	AGD_USR.1: User guidance
ALC: Life cycle support	ALC_DVS.1: Identification of security measures
	ALC_FLR.2: Flaw reporting procedures
	ALC_LCD.1: Developer defined life-cycle model
	ALC_TAT.1: Well-defined development tools
ATE: Tests	ATE_COV.2: Analysis of coverage
	ATE_DPT.1: Testing: high-level design
	ATE_FUN.1: Functional testing
	ATE_IND.2: Independent testing - sample
AVA: Vulnerability assessment	AVA_MSU.2: Validation of analysis
	AVA_SOF.1: Strength of TOE security function evaluation
	AVA_VLA.2: Independent vulnerability analysis

Table 4 EAL 4 augmented with ALC_FLR.2 Assurance Components

5.3.1 Configuration management (ACM)

5.3.1.1 Partial CM automation (ACM_AUT.1)

ACM_AUT.1.1d The developer shall use a CM system.
ACM_AUT.1.2d The developer shall provide a CM plan.
ACM_AUT.1.1c The CM system shall provide an automated means by which only authorized changes are made to the TOE implementation representation.
ACM_AUT.1.2c The CM system shall provide an automated means to support the generation of the TOE.
ACM_AUT.1.3c The CM plan shall describe the automated tools used in the CM system.
ACM_AUT.1.4c The CM plan shall describe how the automated tools are used in the CM system.
ACM_AUT.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.1.2 Generation support and acceptance procedures (ACM_CAP.4)

ACM_CAP.4.1d The developer shall provide a reference for the TOE.
ACM_CAP.4.2d The developer shall use a CM system.
ACM_CAP.4.3d The developer shall provide CM documentation.
ACM_CAP.4.1c The reference for the TOE shall be unique to each version of the TOE.

ACM_CAP.4.2c The TOE shall be labelled with its reference.
ACM_CAP.4.3c The CM documentation shall include a configuration list, a CM plan, and an acceptance plan.
ACM_CAP.4.4c The configuration list shall uniquely identify all configuration items that comprise the TOE.
ACM_CAP.4.5c The configuration list shall describe the configuration items that comprise the TOE.
ACM_CAP.4.6c The CM documentation shall describe the method used to uniquely identify the configuration items that comprise the TOE.
ACM_CAP.4.7c The CM system shall uniquely identify all configuration items that comprise the TOE.
ACM_CAP.4.8c The CM plan shall describe how the CM system is used.
ACM_CAP.4.9c The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.
ACM_CAP.4.10c The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.
ACM_CAP.4.11c The CM system shall provide measures such that only authorised changes are made to the configuration items.
ACM_CAP.4.12c The CM system shall support the generation of the TOE.
ACM_CAP.4.13c The acceptance plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.
ACM_CAP.4.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.1.3 Problem tracking CM coverage (ACM_SCP.2)

ACM_SCP.2.1d The developer shall provide a list of configuration items for the TOE.
ACM_SCP.2.1c The list of configuration items shall include the following: implementation representation; security flaws; and the evaluation evidence required by the assurance components in the ST.
ACM_SCP.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.2 Delivery and operation (ADO)

5.3.2.1 Detection of modification (ADO_DEL.2)

ADO_DEL.2.1d The developer shall document procedures for delivery of the TOE or parts of it to the user.
ADO_DEL.2.2d The developer shall use the delivery procedures.
ADO_DEL.2.1c The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.
ADO_DEL.2.2c The delivery documentation shall describe how the various procedures and technical measures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.
ADO_DEL.2.3c The delivery documentation shall describe how the various procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.
ADO_DEL.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.2.2 Installation, generation, and start-up procedures (ADO_IGS.1)

ADO_IGS.1.1d The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.
ADO_IGS.1.1c The installation, generation and start-up documentation shall describe all the steps necessary for secure installation, generation and start-up of the TOE.
ADO_IGS.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADO_IGS.1.2e The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

5.3.3 Development (ADV)

5.3.3.1 Fully defined external interfaces (ADV_FSP.2)

ADV_FSP.2.1d The developer shall provide a functional specification.
ADV_FSP.2.1c The functional specification shall describe the TSF and its external interfaces using an informal style.
ADV_FSP.2.2c The functional specification shall be internally consistent.
ADV_FSP.2.3c The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing complete details of all effects, exceptions and error messages.
ADV_FSP.2.4c The functional specification shall completely represent the TSF.
ADV_FSP.2.5c The functional specification shall include rationale that the TSF is completely represented.
ADV_FSP.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_FSP.2.2e The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.2 Security enforcing high-level design (ADV_HLD.2)

ADV_HLD.2.1d The developer shall provide the high-level design of the TSF.
ADV_HLD.2.1c The presentation of the high-level design shall be informal.
ADV_HLD.2.2c The high-level design shall be internally consistent.
ADV_HLD.2.3c The high-level design shall describe the structure of the TSF in terms of subsystems.
ADV_HLD.2.4c The high-level design shall describe the security functionality provided by each subsystem of the TSF.
ADV_HLD.2.5c The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.
ADV_HLD.2.6c The high-level design shall identify all interfaces to the subsystems of the TSF.
ADV_HLD.2.7c The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.
ADV_HLD.2.8c The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.
ADV_HLD.2.9c The high-level design shall describe the separation of the TOE into TSP enforcing and other subsystems.
ADV_HLD.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_HLD.2.2e The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.3 Subset of the implementation of the TSF (ADV_IMP.1)

ADV_IMP.1.1d The developer shall provide the implementation representation for a selected subset of the TSF.
ADV_IMP.1.1c The implementation representation shall unambiguously define the TSF to a level of detail such that the TSF can be generated without further design decisions.
ADV_IMP.1.2c The implementation representation shall be internally consistent.
ADV_IMP.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_IMP.1.2e The evaluator shall determine that the least abstract TSF representation provided is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.4 Descriptive low-level design (ADV_LLD.1)

ADV_LLD.1.1d The developer shall provide the low-level design of the TSF.
ADV_LLD.1.1c The presentation of the low-level design shall be informal.
ADV_LLD.1.2c The low-level design shall be internally consistent.
ADV_LLD.1.3c The low-level design shall describe the TSF in terms of modules.
ADV_LLD.1.4c The low-level design shall describe the purpose of each module.

ADV_LLD.1.5c The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.
ADV_LLD.1.6c The low-level design shall describe how each TSP-enforcing function is provided.
ADV_LLD.1.7c The low-level design shall identify all interfaces to the modules of the TSF.
ADV_LLD.1.8c The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.
ADV_LLD.1.9c The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing details of effects, exceptions and error messages, as appropriate.
ADV_LLD.1.10c The low-level design shall describe the separation of the TOE into TSP-enforcing and other modules.
ADV_LLD.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_LLD.1.2e The evaluator shall determine that the low-level design is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.5 Informal correspondence demonstration (ADV_RCR.1)

ADV_RCR.1.1d The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.
ADV_RCR.1.1c For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.
ADV_RCR.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.3.6 Informal TOE security policy model (ADV_SPM.1)

ADV_SPM.1.1d The developer shall provide a TSP model.
ADV_SPM.1.2d The developer shall demonstrate correspondence between the functional specification and the TSP model.
ADV_SPM.1.1c The TSP model shall be informal.
ADV_SPM.1.2c The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modeled.
ADV_SPM.1.3c The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modeled.
ADV_SPM.1.4c The demonstration of correspondence between the TSP model and the functional specification shall show that all of the security functions in the functional specification are consistent and complete with respect to the TSP model.
ADV_SPM.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4 Guidance documents (AGD)

5.3.4.1 Administrator guidance (AGD_ADM.1)

AGD_ADM.1.1d The developer shall provide administrator guidance addressed to system administrative personnel.
AGD_ADM.1.1c The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.
AGD_ADM.1.2c The administrator guidance shall describe how to administer the TOE in a secure manner.
AGD_ADM.1.3c The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.
AGD_ADM.1.4c The administrator guidance shall describe all assumptions regarding user behavior that are relevant to secure operation of the TOE.
AGD_ADM.1.5c The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

AGD_ADM.1.6c The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_ADM.1.7c The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD_ADM.1.8c The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

AGD_ADM.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4.2 User guidance (AGD_USR.1)

AGD_USR.1.1d The developer shall provide user guidance.

AGD_USR.1.1c The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD_USR.1.2c The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD_USR.1.3c The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD_USR.1.4c The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behaviour found in the statement of TOE security environment.

AGD_USR.1.5c The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD_USR.1.6c The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

AGD_USR.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.5 Life cycle support (ALC)

5.3.5.1 Identification of security measures (ALC_DVS.1)

ALC_DVS.1.1d The developer shall produce development security documentation.

ALC_DVS.1.1c The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC_DVS.1.2c The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.

ALC_DVS.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_DVS.1.2e The evaluator shall confirm that the security measures are being applied.

5.3.5.2 Flaw reporting procedures (ALC_FLR.2)

ALC_FLR.2.1d The developer shall provide flaw remediation procedures addressed to TOE developers.

ALC_FLR.2.2d The developer shall establish a procedure for accepting and acting upon all reports of security flaws and requests for corrections to those flaws.

ALC_FLR.2.3d The developer shall provide flaw remediation guidance addressed to TOE users.

ALC_FLR.2.1c The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.2.2c The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.2.3c The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.2.4c The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

ALC_FLR.2.5c The flaw remediation procedures documentation shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.

ALC_FLR.2.6c The procedures for processing reported security flaws shall ensure that any reported flaws are corrected and the correction issued to TOE users.

ALC_FLR.2.7c The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.

ALC_FLR.2.8c The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE.

ALC_FLR.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.5.3 Developer defined life-cycle model (ALC_LCD.1)

ALC_LCD.1.1d The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

ALC_LCD.1.2d The developer shall provide life-cycle definition documentation.

ALC_LCD.1.1c The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.

ALC_LCD.1.2c The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.

ALC_LCD.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.5.4 Well-defined development tools (ALC_TAT.1)

ALC_TAT.1.1d The developer shall identify the development tools being used for the TOE.

ALC_TAT.1.2d The developer shall document the selected implementation-dependent options of the development tools.

ALC_TAT.1.1c All development tools used for implementation shall be well-defined.

ALC_TAT.1.2c The documentation of the development tools shall unambiguously define the meaning of all statements used in the implementation.

ALC_TAT.1.3c The documentation of the development tools shall unambiguously define the meaning of all implementation-dependent options.

ALC_TAT.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.6 Tests (ATE)

5.3.6.1 Analysis of coverage (ATE_COV.2)

ATE_COV.2.1d The developer shall provide an analysis of the test coverage.

ATE_COV.2.1c The analysis of the test coverage shall demonstrate the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.

ATE_COV.2.2c The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.

ATE_COV.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.6.2 Testing: high-level design (ATE_DPT.1)

ATE_DPT.1.1d The developer shall provide the analysis of the depth of testing.

ATE_DPT.1.1c The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design.

ATE_DPT.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.6.3 Functional testing (ATE_FUN.1)

ATE_FUN.1.1d The developer shall test the TSF and document the results.
ATE_FUN.1.2d The developer shall provide test documentation.
ATE_FUN.1.1c The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.
ATE_FUN.1.2c The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.
ATE_FUN.1.3c The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.
ATE_FUN.1.4c The expected test results shall show the anticipated outputs from a successful execution of the tests.
ATE_FUN.1.5c The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.
ATE_FUN.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.6.4 Independent testing - sample (ATE_IND.2)

ATE_IND.2.1d The developer shall provide the TOE for testing.
ATE_IND.2.1c The TOE shall be suitable for testing.
ATE_IND.2.2c The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.
ATE_IND.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ATE_IND.2.2e The evaluator shall test a subset of the TSF as appropriate to confirm that the TOE operates as specified.
ATE_IND.2.3e The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

5.3.7 Vulnerability assessment (AVA)

5.3.7.1 Validation of analysis (AVA_MSU.2)

AVA_MSU.2.1d The developer shall provide guidance documentation.
AVA_MSU.2.2d The developer shall document an analysis of the guidance documentation.
AVA_MSU.2.1c The guidance documentation shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.
AVA_MSU.2.2c The guidance documentation shall be complete, clear, consistent and reasonable.
AVA_MSU.2.3c The guidance documentation shall list all assumptions about the intended environment.
AVA_MSU.2.4c The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).
AVA_MSU.2.5c The analysis documentation shall demonstrate that the guidance documentation is complete.
AVA_MSU.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_MSU.2.2e The evaluator shall repeat all configuration and installation procedures, and other procedures selectively, to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.
AVA_MSU.2.3e The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.
AVA_MSU.2.4e The evaluator shall confirm that the analysis documentation shows that guidance is provided for secure operation in all modes of operation of the TOE.

5.3.7.2 Strength of TOE security function evaluation (AVA_SOF.1)

AVA_SOF.1.1d The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim.

AVA_SOF.1.1c For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.

AVA_SOF.1.2c For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST.

AVA_SOF.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_SOF.1.2e The evaluator shall confirm that the strength claims are correct.

5.3.7.3 Independent vulnerability analysis (AVA_VLA.2)

AVA_VLA.2.1d The developer shall perform a vulnerability analysis.

AVA_VLA.2.2d The developer shall provide vulnerability analysis documentation.

AVA_VLA.2.1c The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for ways in which a user can violate the TSP.

AVA_VLA.2.2c The vulnerability analysis documentation shall describe the disposition of identified vulnerabilities.

AVA_VLA.2.3c The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

AVA_VLA.2.4c The vulnerability analysis documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.

AVA_VLA.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VLA.2.2e The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure the identified vulnerabilities have been addressed.

AVA_VLA.2.3e The evaluator shall perform an independent vulnerability analysis.

AVA_VLA.2.4e The evaluator shall perform independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of additional identified vulnerabilities in the intended environment.

AVA_VLA.2.5e The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker possessing a low attack potential.

6. TOE Summary Specification

This chapter describes the security functions and associated assurance measures.

6.1 TOE Security Functions

6.1.1 Security audit

The ASE Server (or Server) provides its own audit mechanism. Tasks within the Server that represent a client connection can individually cause audit records to be generated. The individual task creates the audit record with the available information relevant to the event. The individual task then places the audit record at the end of the audit queue, which resides in Server shared memory. The Audit Task manages the audit queue. When the Audit Task runs, it removes audit records from the front of the queue and writes them to the 'audit log' which is in the *sybsecurity* database.

The 'audit log', *sysaudits*, consists of one or more tables in the *sybsecurity* database. Using multiple tables for the audit log is extremely useful in the handling of a full audit log condition. The *sysaudits* system table cannot be accessed by any subject other than a System Security Officer; the Server validates this whenever a subject attempts to access the audit log. In addition, the Server does not allow direct delete, update, or insert operations on the audit log. The only way to delete the audit log is to use the truncate table SQL command and only a System Security Officer with appropriate DAC permissions on the audit log can perform this operation.

In general, only the Server writes to the audit log. However, there are times when it may be necessary to allow an 'ad hoc' audit record to be written to the audit log. A special system stored procedure is provided for this purpose. While this stored procedure does not require any roles to execute it, by default, execute permission is not granted to public. Execution of the stored procedure can be allowed or disallowed using the grant/revoke commands, providing a means for administrative control over who can write ad hoc audit records.

The audit log can be read via the SQL select command. The System Security Officer (see section 6.1.5) that installed the audit system may also create views of the audit log or can create stored procedures which include the select command on the audit log. Using the SQL select command, the audit log can be searched and sorted based on any attributes within the audit records, including event type, date, time, success or failure, and user identities.

- All actions that require a System Administrator or System Security Officer role are auditable. Unsuccessful attempts to perform a trusted operation by an untrusted subject also result in the generation of an audit record.

A system stored procedure is provided to control which actions are currently audited. The system stored procedure provides a fine granularity of actions which can be enabled or disabled only by System Security Officer. The stored procedure has the ability to perform the applicable function, but checks the user for appropriate authority before so doing. The specific actions (i.e., event types) which can be enabled and disabled in a desired combination are as follows:

- ad hoc - Allows users to use *sp_addauditrecord*, provided the user has appropriate DAC permission.
- all - Audits all actions. Whenever any auditable action occurs, and audit record is generated. This does not affect the ability to use *sp_addauditrecord*.
- alter - Audits the execution of alter table or alter database
- bcp - Audits the execution of bcp in or bcp out
- bind - Audits the execution of *sp_bindefault*, *sp_bindmsg*, and *sp_bindrule* system procedures
- create - Audits the creation of database objects
- dbaccess - Audits access to the current database from another database
- dbcc - Audits the execution of dbcc commands

- delete - Audits the deletion of rows from a table or view
- disk - Audits the execution of disk init, disk refit, disk reinit, disk mirror, disk unmirror, and disk remirror
- drop - Audits the dropping of database objects
- dump - Audits the execution of dump database or dump transaction commands
- encryption_key – Audits the execution of create, alter and drop encryption key, and the execution of sp_encryption system stored procedure to set system encryption password.
- errors - Audits errors, whether fatal or not
- exec_procedure - Audits the execution of a stored procedure
- exec_trigger - Audits the execution of a trigger
- func_dbaccess - Audits access to a database via TSQL functions
- func_obj_access - Audits access to a database object via a TSQL function
- grant - Audits the execution of the grant command
- insert - Audits the insertion of rows into a table or view
- install - Audits the installation of java classes
- load - Audits the execution of the load database or load transaction commands
- login - Audits all login attempts
- logout - Audits all logouts (intentional or unintentional)
- mount - Audits the execution of mount database command
- quiesce - Audits the execution of quisce database command
- reference - Audits the creation of references between tables
- remove - Audits the removal of java classes
- revoke - Audits the execution of the revoke command
- rpc - Audits the execution of remote procedure calls
- security - Audits the following security-related events: server boot, server shutdown, modification of an audit related configuration parameters, access to audit logs, changing user identity during an established session, role management and role toggling, killing of a user session, regeneration of SSO password and access to objects through TSQL built-in functions.
- select - Audits the execution of the select command
- setuser - Audits execution of setuser command
- table_access - Audits accesses to any table by a specific user or all users
- truncate - Audits the execution of the truncate table command
- unbind - Audits the execution of the *sp_unbindrule*, *sp_unbindmsg*, and *sp_unbindefault* system procedures
- unmount - Audits the execution of unmount database command
- update - Audits updates to rows in a table or view
- view_access - Audits accesses to any view by a specific user or all users

The following information is included in the audit record:

- Event type

- Date/time of event
- Sequence number of record within a single event[3]
- Success/failure in terms of a permission check (if applicable)
- Login ID and name (if applicable)
- Subject's Server ID (SUID)
- Object owner name (if applicable)
- Transaction ID (if applicable)[4]
- Additional information (event based)

A System Security Officer can configure the Audit Mechanism in one of two ways so that audit data is not lost when the current audit log becomes full. The SUSPEND AUDIT WHEN DEVICE FULL audit configuration parameter can be set only by a System Security Officer to one of two values:

- When set to the value one (1), the audit task becomes suspended when the current audit log becomes full. Any user process that generates an auditable event when the audit log is full also becomes suspended. Any suspended processes can only be resumed when a System Security Officer logs in and sets up an empty audit table as the current audit log. Any audit records generated by the System Security Officer during this process are sent to the Server's errorlog.
- When set to the value zero (0), the audit task truncates (i.e., entirely clears) the next audit table and starts using it as the current audit log whenever the current audit log becomes full. In this case, no processes are suspended due to a full audit log condition.

The SUSPEND AUDIT WHEN DEVICE FULL audit configuration parameter is set to one (1) by default. In this configuration, no audit records are lost. If a System Security Officer changes to the value of zero (0), older audit records may be lost if audit tables in the pool of full audit tables are not appropriately backed up.

The Security audit function is designed to satisfy the following security functional requirements:

- FAU_GEN.1: The audit events as well as the audit record content enumerated above represent a superset of that required.
- FAU_GEN.2: A user identity is associated with each audit event that involves a user.
- FAU_SAR.1: The system table containing the audit log is accessible by System Security Officers who can use the SQL select command.
- FAU_SAR.2: The system table containing the audit log is accessible by System Security Officers by default and additional users could be granted read access using the DAC mechanism.
- FAU_SAR.3: The audit log can be searched and sorted based on user identity and other audit records contents using the SQL select command.
- FAU_SEL.1: A system stored procedure can be used to select auditable events to be audited based on audit event type as well as individual users in the case of object access attempts.
- FAU_STG.1: The system table storing the audit log is protected so that only System Security Officers have any access and even though System Security Officers can truncate the audit log they cannot otherwise modify the audit log.
- FAU_STG.3: The TOE offers System Security Officers a parameter that dictates whether to prevent auditable events or discard old audit records when the audit trail becomes full.

[3] In cases where the additional information for an audit record is longer than the space allocated for the additional information column in the record, the information spills into a new row in sysaudits table. The sequence number identifies the sequence in which the rows in sysaudits tables within a single event should be interpreted.
[4] Note that each database transaction is assigned a unique identified so that any related events can be correlated.

6.1.2 Cryptographic support

ASE can be configured with the ASE_ENCRYPTION license option after which it offers the ability to encrypt specific columns of data within a database table. When the license is installed, the System Security Officer can enable and disable the user data encryption function within ASE.

ASE implements three hierarchical layers of encryption in order to protect user data. The first layer involves the use of static data to encrypt a system encryption secret. The second layer involves the use of the system encryption secret to generate a key, called the key encryption key, which is then used to encrypt additional keys, called column encryption keys. The third layer involves the use of the column encryption keys to encrypt specific columns in other databases.

The System Security Officer uses the sp_encryption system stored procedure to enter the system encryption secret. The system encryption secret is specific to the database where sp_encryption is executed, and its encrypted value is stored in the sysattributes system table in that database. The system encryption secret is encrypted using the AES algorithm, CBC mode, and a 128-bit key represented by static data elements in ASE memory. The secret entered by the System Security Officer can be as many as 64 bytes in length and is processed in accordance with KDF1 using SHA-1 in order to generate the key encryption key. The key encryption key is used by ASE to encrypt all column encryption keys stored in the corresponding database.

Once the system encryption secret has been set, it does not have to be supplied again to access keys or data. The column encryption keys are encrypted using the AES algorithm, CBC mode, and the 128-bit key encryption key. Each database that contains column encryption keys would require its own system encryption secret. If all column encryption keys are stored in one designated database, then only that database requires a system encryption secret. Encrypted columns may be created in the same database as the keys or in other databases.

Note that a user could potentially scavenge the data necessary to determine the system encryption secret from the server storage media and, hence, in order to mitigate the potential for subsequent access to user data, the protected user data should be stored in separate databases on different storage media. This would serve to require access to all of the applicable media in order to successfully scavenge applicable protected user data.

Column encryption keys are generated using the ‘create encryption key’ statement, which requires the specification of the key’s name, the encryption algorithm, the key size, the key’s default property, as well as whether an initialization vector or padding is used during the encryption process. Column encryption in ASE uses the Advanced Encryption Standard (AES) symmetric key encryption algorithm, with available key sizes of 128, 192, and 256 bits. Random-key generation (and cryptographic functionality) is provided by the Security Builder Crypto API in accordance with ANSI X9.62.

The System Security Officer can create (as indicated above) column encryption keys directly and/or grant create encryption key (i.e., CREATE ENCRYPTION KEY) permission to other users in a database. Separate encryption keys can be created for each column to be encrypted. Encryption keys can be shared between columns, but each column can have only one encryption key. The System Security Officer can set up a default encryption key for a given database. The default encryption key is used whenever the encrypt qualifier is used without a key name on create table, alter table, and select into. The encryption key owner must grant ‘select’ permission on encryption keys to users executing create/alter table commands before the encryption key can be used to encrypt database columns.

Columns in new tables can be marked for encryption using the ‘encrypt’ option in ‘create table’ command. Columns in existing tables can be modified and marked for encryption using the ‘encrypt’ option in ‘alter table’ command. In either case, the contents of the column will be encrypted using the AES algorithm, CBC or ECB mode (depending on whether the initialization vector is specified), using the column encryption key (128, 192, or 256 bits) specified by the user using create or alter table statement or the default database encryption key if no key is specified (provided the user has select permission to the applicable key as indicated above). Once a column is encrypted, in order to access that column the applicable user must either have decrypt access on the table (which implies decrypt permission on all columns in that table) or decrypt access on the applicable columns. With the applicable permissions, data in the column is encrypted automatically when written and decrypted automatically when read using standard SQL data access statements. The column encryption keys required for encryption and decryption operations are automatically obtained from *sysencryptkeys* system table and decrypted in memory for the duration of the cryptographic operation. Once the cryptographic operation is complete, the in-memory copy of cleartext key is destroyed by zeroing out the memory.

Table owners can drop encryption on columns using the 'decrypt' option in 'alter table' statements. The System Security Officer and key owners can drop keys using the 'drop encryption key' command. A key can be successfully dropped only if there are no encrypted columns in any database that use the key.

When the encryption function is disabled at the system level, any attempts to access any columns that have been previously encrypted fail with an error message and no additional columns can be encrypted. When the function is re-enabled, access to any previously encrypted columns is restored and new columns can once again be encrypted.

Note that when an existing column is encrypted, all of its contents are encrypted at that point and replace the previously unencrypted content. Alternately, when a column is decrypted, the encrypted contents are decrypted and replace the previously encrypted content. In order to turn off decryption at the system level and leave all of the data accessible (albeit no longer encrypted), each encrypted column must be first decrypted and then the system level setting can be configured to disable encryption.

ASE destroys raw keys in memory by zeroizing out the memory allocations containing the raw key. Additionally, ASE calls the Certicom library functions to destroy in-memory key references as soon as cryptographic operations are completed. Keys are stored encrypted on disk and may be destroyed (i.e., deleted) by a System Security Officer if they are not being used to encrypt user data using the 'drop encryption key' command.

The Cryptographic support function is designed to satisfy the following security functional requirements:

- FCS_CKM.1: ASE generates keys for use in the symmetric encryption and encryption of the contents of encrypted database columns.
- FCS_CKM.4: ASE destroys raw keys in memory by zeroing out the memory locations containing the raw keys. When directed by a System Security Officer, ASE deletes keys from disk.
- FCS_COP.1a: ASE encrypts and decrypts data written to and read from, respectively, encrypted database columns.
- FCS_COP.1b: ASE hashes the system encryption secret in support of KDF key generation to generate key encryption keys.

6.1.3 User data protection

The ASE Server provides the ability to allow access to be controlled through a combination of two policies: Discretionary Access Control (DAC) and Policy-based Access Control.

6.1.3.1 Discretionary Access Policy

The DAC policy for object access is based on:

- user identities,
- group memberships (including 'public'),
- object ownership,
- active roles, and
- Access Control Lists (ACLs).

The ASE Server objects directly subject to this policy are databases, tables, views, and stored procedures. Other objects are always public (e.g., global variables, messages, defaults and rules), always private (e.g., temporary tables), or are part of one of the protected objects (e.g., indexes and triggers, are protected via the tables with which they are associated).

The Server stores the access permissions for the applicable objects in access control lists (ACLs) and provides the ability to grant and revoke ANSI Standard SQL permissions. The user identity (specifically, the combination of SUID and UID - as identified in the Identification and Authentication function discussion below), group memberships and active roles for the user identity, are used by the DAC mechanism to validate the user's permission to access the applicable objects.

Initially, only a System Administrator can create a database. The System Administrator can grant, and subsequently revoke, the permission to create databases to other users. When a user creates a database, that user becomes the Database Owner (DBO) and inherits the ability to create, drop and (where relevant) alter, the following database objects: tables, indexes, procedural objects (stored procedures, triggers and views), defaults, and rules. To access anything in a database, a user must have use access for the particular database which can be granted only by the DBO or System Administrator via a system stored procedure. The stored procedure adds an entry to the *sysusers* system table (i.e., the ACL) allowing the user to access the particular database.

The DBO or System Administrator can subsequently grant users the permission to create tables, views, and stored procedures, as well as other capabilities in the database. For tables and views, the grantable permissions are insert, select, update, delete, alter (tables only), truncate table (tables only), update statistics (tables only), delete statistics (tables only), and references (tables only). When a user creates a table, the user becomes the table owner and inherits the following permissions: alter table, create/drop index, create/drop trigger, truncate table, update/delete statistics, bind defaults and rules to the columns of the table.

If a user has DAC permission on a view, the user also has implied DAC permission on all objects upon which the view depends (via the view only), provided the owner of the view also owns all of the objects upon which the view depends.

For stored procedures the only grantable permission is execute. If a user has execute permission on a stored procedure, the user can access all objects referenced by the stored procedure provided they are all owned by the owner of the stored procedure. If an object referenced in a procedural object (stored procedure, view, trigger) is not owned by the owner of the procedural object, the user must have the necessary access permission on the referenced object.

Permissions can be granted on columns, which are seen as part of a table, and include select, references, and update. For example, select permission can be granted on an entire table or just specific columns of a table.

Permissions can be granted with or without the grant option. The grant option permits the user to subsequently grant the specific permission to other users.

Permissions can be granted on encryption keys stored within a database. Only the creator of an encryption key can grant select permission so that the key can be used by other users when encrypting database columns.

Note that an ACL can also explicitly revoke (i.e., deny) specific access permissions for specified users, groups, or roles. In this case, the specified user or a user with the specified group or role cannot use the associated access permissions, regardless of other permissions that may be granted via the ACL.

6.1.3.1.1 Users and Groups

To access anything in a database, a user must have use access for the particular database which can be granted only by the DBO or System Administrator via a system stored procedure. The stored procedure adds an entry to the *sysusers* system table allowing the user to access the particular database. If the user does not have an explicit entry in *sysusers*, the user can still access the database if a 'guest' entry is present provided the user has a valid, unlocked account in master database system table.

While user identities can be used in ACLs to assign specific access permissions to specific users, ASE also supports a 'group' and a 'roles' feature.

Groups provide a convenient way to grant and revoke permissions to more than one user in a single statement. Every database user is implicitly a member of the special group 'public'. The group 'public' can never be dropped. A user can also be a member of at most one additional group; membership in roles is not affected by this limit. Note, however, that roles can be used like groups in assigning access to objects. The following DAC permissions are always available to 'public':

- begin transaction
- commit
- create <temporary table>
- print

- raiserror
- rollback
- save transaction
- set

User groups are specific to a database and can only be created by a System Administrator, System Security Officer, or the DBO (for their own database) using the system stored procedure *sp_addgroup* (which restricts the use of the procedure accordingly). Group information is also stored in a system table and in addition to the access permissions an ACL might grant to a user, the user can also access operations granted to any of their assigned groups.

6.1.3.1.2 Ownership and DAC Permissions

The owner of a database is referred to as the DBO and owns all of the system tables and system stored procedures in the database when the database is first created. However, no user (neither the DBO nor any other security management role) can directly update the system tables.

The DBO has all command permissions on the database, some of which may be granted to database users, as follows:

Permissions Grantable by DBO

- create default
- create procedure
- create rule
- create table (implies create trigger, create index)
- create view
- set proxy

Permissions Not Grantable by DBO

- alter database
- checkpoint
- dbcc
- dump/load [database or transaction]
- revoke <command>
- setuser
- drop database

NOTE: Check whether the above table is complete? How about “create encryption key” permission?

When a user in the database creates an object, that user becomes the object owner. For example, if user Joe creates table1, then Joe is the table owner (TBO) of table1. Object owners in a database have permissions, some of which may be granted to other database users, as follows:

Permissions Grantable by Object Owners

- delete
- execute
- grant <object>
- insert
- readtext, writetext
- references
- select
- update
- truncate table
- update statistics
- delete statistics
- decrypt

Permissions Not Grantable by Object Owners

- alter table
- create index
- create trigger
- drop <object>
- revoke <object>

Note that the commands listed in the table apply to different objects, for example, execute only applies to stored procedures. Table DAC permissions (insert, delete, update, select, references, truncate table, update statistics, delete statistics, decrypt) apply to all columns in a table by default. Update, select, references, and decrypt may be restricted to a subset of columns in a table.

6.1.3.2 Policy-based Access Control

The Policy-based Access Control policy is implemented through the combined capabilities of:

- Access Rules that the DBO defines and binds to the table,
- Application Context Facility, which provides built-in functions that define, store, and retrieve user-defined contexts, and
- Login triggers that the Database Owner or the user can create.

6.1.3.2.1 Access Rules

Access Rules are the fundamental building blocks of Policy-Based Access Control. An Access Rule is bound to a specific column and then invoked on any select, update, or delete operation on the corresponding table. So, while the rule is applied to a given column, the rule effectively determines which rows in the given table are accessible. Furthermore, if a given row is not accessible due to a rule on a given column, that row will not be returned at all.

6.1.3.2.2 Application Context Facility

ASE's Application Context Facility supports application-specific security policies that are enforced in the server itself. This means that they are always enforced and can be changed and updated without any change to the application. An Application Context consists of a series of (context, attribute, value) tuples. "Context" is the name of a user-defined context that can be used by one or more applications. You can have a separate security context for each application, or you can use the same security context for a set of applications with similar access control requirements. "Attributes" are the variables that are used in Access Rules, and an attribute will be assigned a specific "value" when a context is being set up. Contexts are set up on a session-by-session basis, so they allow the security policy to be based on properties of both the application and the user invoking the application.

For example, a sales reporting application could utilize a context that sets values for Region and Manager attributes.

- (Sales_Context, Region, "North America")
- (Sales_Context, Manager, "TRUE")

Access Rules could use this context to support a security policy that states that sales representatives can only view information about their own accounts, but managers can view all account information within their region.

6.1.3.2.3 Login Triggers

Login triggers are stored procedures that execute as part of the login process. Login triggers simplify the ongoing administration of security policies in ASE. They are a convenient way of configuring an Application Context by looking up tables with organizational and entitlement information, and setting values for all of the attributes within a context. In the sales reporting example, an HR table could be referenced to set the Manager attribute. When an employee's Manager status changes, the security policy applied to that employee would be updated automatically and would be applied the next time that employee attempts to log in.

In addition, login triggers can query other tables and use that data to support a number of different account usage policies. For example, a login trigger could be used to prevent users from logging in outside of regular business hours or to lock users' accounts during vacation.

6.1.3.3 Residual Information Protection

When a database is dropped, all rows in the system tables of the master database which reference or define the database are deleted. The actual data pages are not zeroed out, but they are not accessible to other databases. The data pages are zeroed out when they are allocated to a newly created database.

When a table or index is created, pages are allocated from the next 8-page extent in the database. Although the data areas of these pages are not zeroed out before use, the page header information is updated whenever a new page is used for the object. The information in the database's allocation pages, the allocation map for the object, and the page headers, ensures that only data which has been written out may be accessed.

When a table is dropped, all rows in the system tables of the database in which the table resides which reference the table and its associated indexes are deleted. The allocation bitmaps for all extents allocated to the table and its

indexes are zeroed out so that there is no access to those pages for that object ID (Tables, views, rules, and defaults have an object ID). Object IDs are never reused, and are guaranteed to be unique. Truncating a table has the effect of deleting all data rows for the table and de-allocating the associated data pages and extents from the table.

The Server expects files and raw device partitions used to store configuration information and data to be effectively clear or empty when they are obtained from the underlying operating system. However, as indicated above, the Server is designed to initialize or otherwise ensure that reads are not allowed before writes which alleviates dependency on the operating system to actually clear the required resources. Eventually, when a file or device partition is no longer necessary, the Server will effectively release it. Such files and device partitions are not cleared by the Server, but it is expected that the operating system would not allow any residual contents to become available to some other application (e.g., by reformatting or some other mechanism).

The Server expects the shared memory segments provided by the underlying operating system to be clear when they are created. However, once the shared memory segments are obtained, they are not released by the Server until it shuts down at which time it is expected that the operating system would not allow any residual contents to become available to some other application. The Server uses these shared memory segments to store global data structures and buffers. The Server object reuse policy on shared memory segments is either write-before-read or initialize when the structure or buffer is created, alleviating dependency on the operating system to actually clear the resources granted to the Server.

The User data protection function is designed to satisfy the following security functional requirements:

- FDP_ACC.1a: All database subjects are subject to the DAC policy for all available operations on databases, tables, views, and stored procedures (and their contents).
- FDP_ACC.1b: All database subjects are subject to the Policy-based Access Control policy when selecting, updating, or deleting data within database tables.
- FDP_ACF.1a: Database objects have owners. Object ownership and ACLs are compared against user identities and group memberships for that user in order to determine whether the request operation should be allowed. Alternately, a user may have an active role that explicitly grants the requested access regardless of the normal access check. If both of these checks fail, access will be refused. Note that both the DAC and Policy-based Access Control policies must be simultaneously satisfied in order to obtain access.
- FDP_ACF.1b: Application contexts are established using Login Triggers and are subsequently compared against Access Rules associated with columns within database tables when access is attempted. Access will be allowed only if there is no Access Rule or if the Access Rule allows the access. Note that both the DAC and Policy-based Access Control policies must be simultaneously satisfied in order to obtain access.
- FDP_RIP.2a: Objects are cleared when allocated, either by zeroing the data structures or by overwriting the data structures with their new contents prior to being accessible.

6.1.4 Identification and authentication

The ASE Server supports an identification and authentication mechanism in addition to that of the underlying operating system. In order to access the Server, a login account, including a login name and password, must be created for the user. User accounts can optionally be associated with a trusted role (e.g., System Administrator), as described in relation to the Security Management function (below). Login name, password and an internal Server identifier are stored in the *syslogins* system table in the master database. The select permission is granted to ‘public’ but revoked for the password and *audflags* columns of *syslogins*. Note that users can also be assigned to groups. Such assignments are specific to each database and can be made only by a System Administrator., System Security Officer, or DBO (in the context of their own database). The group information is associated with each user session when a connection is made to a specific database.

To login to the Server, the user provides the login name and password to the Server. The Server compares the password to that stored in the *syslogins* system table. If either login name or password is incorrect the login request will fail and no TSF-mediated functions will be made available.

A System Security Officer can define a maximum number of failed login attempts for each user. If this number is exceeded, the associated user will no longer be able to successfully login until a System Security Officer re-enables

the user account. Similarly, the System Security Officer is provided guidance in the administrator guide to define restrictions (e.g., minimum password length) that are enforceable by ASE Server to ensure that the chance of guessing a password is sufficiently small (i.e., less than one in 500,000,000,000 – based on a default minimum password length of 6, a password alphabet of 94 characters, and appropriate guidance for the selection of passwords). Note that the login interface does not echo passwords back to the user and when a failure occurs, the TOE indicates only that the identity or password was incorrect.

As a result of a successful login, a subject is created on behalf of the client user and is represented by a user identity which is represented by the combination of a unique Server-wide identifier (SUID) and a database-specific identifier (UID). The subject is actually a separate task instantiated within the ASE Server assigned to the user's TDS connection and associated with the user's identifiers. Once a session is established, ASE provides a number of commands to change the identity of the session. A System Security Officer can freely change their identity and can also explicitly grant other users the ability to change their identities. When assigning the ability to change identity, the System Security Officer can specify a user list and a set of roles that serve to restrict the identities that can be assumed via this ability (note that the set of roles restricts the assumable identities to those that share the identified roles). Furthermore, while all of a users system defined roles and the roles added as default roles are enabled when a session is established, a user can freely enable and disable any of their assigned roles during the session.

In addition to the user's identifiers and password, any roles assigned to the user are stored in the system tables. The set of active roles in a given user session is stored in in-memory session specific data structures. Note that roles represent sets of privileges available to the user.

The Identification and authentication function is designed to satisfy the following security functional requirements:

- FIA_AFL.1: A System Security Officer can define the maximum number of failed login attempts before the user account becomes disabled until re-enabled by a System Security Officer.
- FIA_ATD.1: The TOE defines user identities, authentication data, and roles within system tables.
- FIA_SOS.1: The System Security Officer is provided guidance necessary to configure the authentication parameters (e.g., minimum password length) necessary to comply with this requirement.
- FIA_UAU.2: The TOE offers no TSF-mediated functions until the user is authenticated.
- FIA_UAU.7: The TOE does not echo passwords as they are entered and does not identify whether applicable authentication failures are due to a bad identity, password, or some other reason.
- FIA_UID.2: The TOE offers no TSF-mediated functions until the user is identified.
- FIA_USB.1: Each user is identified and authenticated when a connection is made to the TOE and the TOE instantiates a task with the user attributes for the duration of the connection. Changes to user attributes associated with a connection are limited to changing identity and active roles – identities can be changed only by a System Security Officer or a user that has been granted that ability by a System Security Officer and only roles from the set assigned to the applicable user can be freely enabled and disabled.

6.1.5 Security management

ASE maintains a number of special tables to control how it operates. Some system tables apply to the entire ASE instance - these are referred to as master database system tables. Other tables apply only to specific database instances - these are referred to simply as system tables. All system tables are protected, using privileges, such that only the appropriately authorized users (System Administrator, System Security Officer, DBO) can update their contents, though select access is publicly offered for some contents when appropriate.

Note that in general updates to system tables are done as effects of SQL commands such as create table or via system stored procedures. Each of these commands is controlled using the DAC mechanism.

Among other things, the system tables are responsible to define user accounts including authentication data (i.e., passwords), associate system-defined roles (e.g., System Administrator, System Security Officer, and Operator) with user accounts, enable and disable encryption, define an encryption password, define a default database encryption key, define column encryption keys, define audit parameters, store audit events, and define databases. As a result, all of the related management functions are restricted to appropriate administrator role.

Each of the system tables can be managed by an appropriate administrator role using the utility provided for that purpose: *isql*. Any attempts to modify those tables will be accepted only if the data provided is valid within those tables. Furthermore, any changes made within those tables will be effective the next time the data within the table is accessed (e.g., a new user connection).

Access to other database objects, such as databases and their contents is subject to the Discretionary Access Control Policy and any user with sufficient privilege, according to that policy, can manage the associated access attributes. Like the system table data changes, changes to access attributes of database objects will be used during the next attempt to access that object. Any database object that is created is initially accessible only by the creator who can subsequently change the access permissions at their discretion.

The Security management function is designed to satisfy the following security functional requirements:

- FMT_MOF.1: The ability to enable and disable the audit and cryptographic related functions is restricted to a System Security Officer through access controls on the applicable system tables and system stored procedures.
- FMT_MSA.1: The ability to manage database subject attributes is restricted to a System Administrator, System Security Officer, or DBO (with regard to assigning group memberships within the context of their own database) through access controls on the applicable master database system tables, SQL commands and system stored procedures.
- FMT_MSA.2: ASE rejects invalid data and given the flexible nature of each of the security functions in operating properly with the data that is provided, the only ‘insecure’ data that could be entered by an administrator would effectively be data that is invalid for the associated function. Hence, ASE prevents the introduction of 'insecure' data.
- FMT_MSA.3: By default every database object is created with the creator as the owner. Subsequently, access can be granted to other users, but there is no method to specify access other than the default during creation.
- FMT_MTD.1a: The ability to configure audited events is restricted to System Security Officers through access controls on the applicable system tables and system stored procedures.
- FMT_MTD.1b: The ability to delete or review audited events is restricted to System Security Officers through access controls on the applicable system tables.
- FMT_MTD.1c: The ability to set and reset subject authentication data is restricted to a System Security Officer and users (who can change only their own user authentication data) through access controls on the applicable system tables and system stored procedures.
- FMT_MTD.1d: The ability to set and reset the encryption password and default database encryption key is restricted to System Security Officers through access controls on the applicable system tables and system stored procedures.
- FMT_REV.1a: The ability to manage database subject attributes is restricted to System Administrators, System Security Officers, and Database Owners (who can set group assignments that affect their own database only) through access controls on the applicable master database system tables and system stored procedures. These tables are used to determine subject attributes each time a user connects to the ASE Server.
- FMT_REV.1b: The ability to manage database object attributes is limited to System Administrators and users based upon the users access to the database object. The owner of an object can always manipulate its attributes, as can the System Administrator. Other users can do so only went the applicable privileges have been granted.
- FMT_SMF.1: A utility is available providing administrators with the interface necessary to perform all management functions, including: start and stop the audit function, select the auditable events to be audited, review the audit data, enable and disable the cryptographic function, manage the encryption password and default database encryption key, creation of column encryption keys, and manage database subjects including authentication data.

- FMT_SMR.1: Each user account can be assigned zero or more system-defined roles. The TOE implements at least the System Administrator and System Security Offifer system-defined roles along with DBO and users (who have responsibility for their own accounts and data).

6.1.6 Protection of the TSF

ASE instantiates itself as a process within task constructs provided by the underlying operating system. It retains exclusive control of its processes and separates and differentiates client connections based on TDS connections. Each such connection is handled by a distinct task within the ASE Server process. In addition to protecting its own processes, ASE protects its shared memory and files using features provided by the underlying operating system. Specifically, it ensures that the security properties of those objects do not allow access by other operating system processes. This serves to both protect ASE itself as well as to ensure that any attempts to access the database constructs realized by ASE must be made through ASE. Furthermore, ASE has been carefully designed to offer well-defined interfaces that ensure that access to protected resources is subject to the applicable ASE security policies.

The Protection of the TSF function is designed to satisfy the following security functional requirements:

- FPT_RVM.1a: The TOE uses protected media (disk and memory) to store the objects it instantiates to ensure that any access attempts must go through ASE where the appropriate access rules are enforced.
- FPT_SEP.1a: The TOE instantiates itself as a process which it protects from inappropriate access. The TOE separates clients based on individual protocol connections.

6.1.7 Resource utilization

ASE allows administrators to define the following limits on queries and transactions issued by database subjects[5]:

- Maximum I/O cost,
- Maximum number of rows that can be returned,
- Maximum number of pages used in a temporary database, and
- Maximum elapsed time to process.

The administrator can also elect to define specific time periods for these restrictions so that, for example, resources can be limited only during times of peak activity. These limits are configured and enforced using system stored procedures provided by ASE.

If resource limits are enabled for the server, any changes to resource limits are immediately effective for all current users. If resource limits are not enabled for the server, a System Administrator is notified of this when attempting to change resource limits and configured limits become effective immediately when resource limits are enabled on the server.

The Resource utilization function is designed to satisfy the following security functional requirements:

- FRU_RSA.1: Stored procedures allow administrators to define query and transaction limits to prevent excessive use by any single database subject.

6.1.8 TOE access

ASE allows System Security Officers to define 'Login Triggers' which are stored procedures that are activated each time a client logs into the ASE Server. Login triggers can effectively deny access to the ASE Server based on criteria defined by the System Security Officer. Among the criteria that can be configured are disallowed time

[5] Note that the limits apply to individual queries and transactions as issued via client connections that represent users (i.e., database subjects). As such, the quotas do not generally apply to users since a given user could establish multiple concurrent client connections.

periods and the maximum number of current sessions the user has. When the login trigger is activated, it checks the applicable attributes against the defined criteria and if any of the rules match, the connection is rejected.

Furthermore, the System Security Officer can directly lock and unlock a user account using a stored procedure.

The TOE access function is designed to satisfy the following security functional requirements:

- FTA_MCS_EXP.1: Login Triggers can be used to restrict the number of concurrent client sessions.
- FTA_TSE.1: Login Triggers can be used to restrict specific user sessions based on the current time and a store procedure is available to restrict user access by locking their account.

6.2 TOE Security Assurance Measures

6.2.1 Configuration management

The configuration management measures applied by Sybase ensure that configuration items are uniquely identified, and that documented procedures are used to control and track changes that are made to the TOE. Sybase ensures changes to the implementation representation are controlled with the support of automated tools and that TOE associated configuration item modifications are properly controlled. Sybase performs configuration management on the TOE implementation representation, design documentation, tests and test documentation, user and administrator guidance, delivery and operation documentation, life-cycle documentation, vulnerability analysis documentation, configuration management documentation, and security flaws.

These activities are documented in:

- Sybase Adaptive Server Enterprise Configuration Management Plan

The Configuration management assurance measure satisfies the following EAL 4 augmented with ALC_FLR.2 assurance requirements:

- ACM_AUT.1
- ACM_CAP.4
- ACM_SCP.2

6.2.2 Delivery and operation

Sybase provides delivery documentation and procedures to identify the TOE, allow detection of unauthorized modifications of the TOE and installation and generation instructions at start-up. Sybase's delivery procedures describe all applicable procedures to be used to detect modification to the TOE. Sybase also provides documentation that describes the steps necessary to install Adaptive Server Enterprise in accordance with the evaluated configuration.

These activities are documented in:

- Supplement for Installing Adaptive Server for Common Criteria Configuration, Document ID: DC00080-01-1252-01
- Sybase Adaptive Server Enterprise Delivery and Operation Procedures, v 0.1, 04/30/2004
- Installation Guide Adaptive Server Enterprise for Digital UNIX
- Installation Guide Adaptive Server Enterprise for HP-UX
- Installation Guide Adaptive Server Enterprise for IBM RISC System/6000 AIX
- Installation Guide Adaptive Server Enterprise for Linux/Intel
- Installation Guide Adaptive Server Enterprise for Silicon Graphics IRIX
- Installation Guide Adaptive Server Enterprise for Sun Solaris

- Installation Guide Adaptive Server Enterprise for Windows NT
- Supplement for Installing Adaptive Server for Common Criteria Configuration

The Delivery and operation assurance measure satisfies the following EAL 4 augmented with ALC_FLR.2 assurance requirements:

- ADO_DEL.2
- ADO_IGS.1

6.2.3 Development

Sybase has numerous documents describing all facets of the design of the TOE. In particular, they have a functional specification that describes the accessible TOE interfaces; a high-level design that decomposes the TOE architecture into subsystems and describes each subsystem and their interfaces; a low-level design that further decomposes the TOE architecture into modules and describes each module and their interfaces; and, correspondence documentation that explains how each of the design abstractions correspond from the TOE summary specification in the Security Target to the actual implementation of the TOE. Furthermore, Sybase has a security model that describes each of the security policies implemented by Adaptive Server Enterprise. Of course, the implementation of the TOE itself is also available as necessary.

These activities are documented in:

- Object Reuse Design Specification
- Object Reuse Prevention Functional Specification
- Discretionary Access Control Functional Specification
- Access Control List Management Design Specification
- Policy Based / Row Level Access Control Functional Specification
- Policy Based / Row Level Access Control Design Specification
- Groups and System Defined Roles Functional Specification
- Groups and System Defined Roles Design Specification
- User Defined Roles Functional Specification
- User Defined Roles Design Specification
- Resource Governor Functional Specification
- Resource Governor Design Specification
- ASE Self Protection Functional Specification
- ASE Self Protection Design Specification
- Adaptive Server Enterprise Architecture Summary
- Adaptive Server Enterprise Auditing Functional Specification
- Adaptive Server Enterprise Auditing Design Specification
- Identification and Authentication Design Specification
- Identification and Authentication Functional Specification
- Security Management Functions Functional Specification
- Encrypted Columns Functional Specification
- Encrypted Columns Design Specification

- Configuration Interface Functional Specification
- Configuration Interface Design Specification
- Reference Validation Mechanism Design Specification
- Dynamic Reconfiguration Design Specification
- Dynamic Reconfiguration Functional Specification
- TDS 5.0 Functional Specification, Version 3.6
- ISQL Functional Specification
- T-SQL Correspondence
- Adaptive Server Enterprise Security Policy Model
- Source code (sample)

The Development assurance measure satisfies the following EAL 4 augmented with ALC_FLR.2 assurance requirements:

- ADV_FSP.2
- ADV_HLD.2
- ADV_IMP.1
- ADV_LLD.1
- ADV_RCR.1
- ADV_SPM.1

6.2.4 Guidance documents

Sybase provides administrator and user guidance on how to utilize the TOE security functions and warnings to administrators and users about actions that can compromise the security of the TOE.

These activities are documented in:

- Common Criteria Evaluation Road Map
- Platform specific Release bulletin(s) for Adaptive Server 15.0.1
- Configuration Guide Adaptive Server Enterprise for UNIX
- Configuration Guide Adaptive Server Enterprise for Windows NT
- Sybase ASE 15.0 System Administration Guide
- Sybase ASE 15.0 Reference Manual: Commands
- Sybase ASE 15.0 Reference Manual: Procedures
- Sybase ASE 15.0.1 Using Encrypted Columns in Adaptive Server

The Guidance documents assurance measure satisfies the following EAL 4 augmented with ALC_FLR.2 assurance requirements:

- AGD_ADM.1
- AGD_USR.1

6.2.5 Life cycle support

Sybase ensures the adequacy of the procedures used during the development and maintenance of the TOE through the use of a comprehensive life-cycle management plan. Sybase includes security controls on the development environment that are adequate to provide the confidentiality and integrity of the TOE design and implementation that is necessary to ensure the secure operation of the TOE. Sybase achieves this through the use of a documented model of the TOE life cycle and well-defined development tools that yield consistent and predictable results. In addition, Sybase identifies and tracks reported flaws, ensuring that they are addressed and corrections and corrective measures are made available as applicable.

These activities are documented in:

- Sybase Adaptive Server Enterprise Life Cycle Document

The Life cycle support assurance measure satisfies the following EAL 4 augmented with ALC_FLR.2 assurance requirements:

- ALC_DVS.1
- ALC_FLR.2
- ALC_LCD.1
- ALC_TAT.1

6.2.6 Tests

Sybase has a test plan that describes how each of the necessary security functions is tested, along with the expected test results. Sybase has documented each test as well as an analysis of test coverage and depth demonstrating that the security aspects of the design evident from the functional specification and high-level design are appropriately tested. Actual test results are created on a regular basis to demonstrate that the tests have been applied and that the TOE operates as designed.

These activities are documented in:

- Sybase ASE Common Criteria Test Plan, v4.0
- Test Suite Documents and associated tests
    - Identification and Authentication
    - Row Level Access Control
    - Auditing
    - Discretionary Access Control (DAC)
    - Resource Governor
    - Dynamic Reconfiguration
    - Groups and System Defined Roles
    - User Defined Roles
    - Configuration Interfaces
    - isql (Interactive SQL parser)
    - ASE Self Protection
    - Security Management Functions
    - Object Reuse Prevention
    - Tabular Data Stream (TDS)
- Test Mapping to Design Specifications

- T-SQL Correspondence
- Actual Test Results for all OS platforms

The Tests assurance measure satisfies the following EAL 4 augmented with ALC_FLR.2 assurance requirements:

- ATE_COV.2
- ATE_DPT.1
- ATE_FUN.1
- ATE_IND.2

6.2.7 Vulnerability assessment

The TOE administrator and user guidance documents describe the operation of Adaptive Server Enterprise and how to maintain a secure state. These guides also describe all necessary operating assumptions and security requirements outside the scope of control of the TOE. They have been developed to serve as complete, clear, consistent, and reasonable administrator and user references. Furthermore, Sybase has conducted a misuse analysis demonstrating that the provided guidance is complete.

Sybase has conducted a strength of function analysis wherein all permutational or probabilistic security mechanisms have been identified and analyzed resulting in a demonstration that all of the relevant mechanisms fulfill the minimum strength of function claim, SOF-Medium.

Sybase performs regular vulnerability analyses of the entire TOE (including documentation) to identify weaknesses that can be exploited in the TOE.

These activities are documented in:

- Sybase ASE Vulnerability Analysis

The Vulnerability assessment assurance measure satisfies the following EAL 4 augmented with ALC_FLR.2 assurance requirements:

- AVA_MSU.2
- AVA_SOF.1
- AVA_VLA.2

7. Protection Profile Claims

There are no Protection Profile claims.

8. Rationale

This section provides the rationale for completeness and consistency of the Security Target. The rationale addresses the following areas:

- Security Objectives;
- Security Functional Requirements;
- Security Assurance Requirements;
- Strength of Functions;
- Security Functional Requirement Dependencies;
- Explicitly Stated Requirements;
- TOE Summary Specification; and
- PP Claims.

8.1 Security Objectives Rationale

This section shows that all secure usage assumptions, organizational security policies, and threats are completely covered by security objectives. In addition, each objective counters or addresses at least one assumption, organizational security policy, or threat.

8.1.1 Security Objectives Rationale for the TOE and Environment

This section provides evidence demonstrating the coverage of organizational policies and usage assumptions by the security objectives.

	P.ACCOUNTABILITY	P.AUTHORIZATION	P.AUTHORIZED_USERS	P.I_AND_A	P.NEED_TO_KNOW	P.ROLES	T.ADMIN_ERROR	T.AUDIT_COMPROMISE	T.MASQUERADE	T.RAW_ACCESS	T.RESIDUAL_DATA	T.RESOURCE	T.SYSACC	T.TSF_COMPROMISE	T.UNAUTH_ACCESS	T.UNDETECTED_ACTIONS	T.UNIDENTIFIED_ACTIONS	A.NO_EVIL	A.NO_GENERAL_PURPOSE	A.PHYSICAL	A.ROBUST_ENVIRONMENT	A.NETWORK
O.ACCESS		X	X		X								X		X							
O.ADMIN_ROLE						X																
O.AUDIT_GENERATION	X							X								X						
O.AUDIT_PROTECTION								X								X						
O.AUDIT_REVIEW	X																X					
O.CRYPTO										X												
O.DISCRETIONARY_ACCESS					X										X							
O.INTERNAL_TOE_DOMAINS															X							
O.MANAGE							X						X				X					
O.PROTECT		X			X										X							
O.RESIDUAL_INFORMATION											X											
O.RESOURCE												X										
O.TOE_PROTECTION														X								
O.USER_AUTHENTICATION				X					X				X									

	P.ACCOUNTABILITY	P.AUTHORIZATION	P.AUTHORIZED_USERS	P.I_AND_A	P.NEED_TO_KNOW	P.ROLES	T.ADMIN_ERROR	T.AUDIT_COMPROMISE	T.MASQUERADE	T.RAW_ACCESS	T.RESIDUAL_DATA	T.RESOURCE	T.SYSACC	T.TSF_COMPROMISE	T.UNAUTH_ACCESS	T.UNDETECTED_ACTIONS	T.UNIDENTIFIED_ACTIONS	A.NO_EVIL	A.NO_GENERAL_PURPOSE	A.PHYSICAL	A.ROBUST_ENVIRONMENT	A.NETWORK
O.USER_IDENTIFICATION	X	X		X	X				X				X									
OE.TIME	X															X						
OE.TOE_PROTECTION														X								
OE.CONFIG																		X				
OE.NO_GENERAL_PURPOSE																			X			
OE.PHYSICAL								X					X	X	X	X				X		
OE.TRUST_IT																					X	
OE.NETWORK																						X

Table 5 Environment to Objective Correspondence

8.1.1.1 P.ACCOUNTABILITY

The users of the TOE shall be held accountable for their actions within the TOE.

This Organizational Policy is satisfied by ensuring that:

- O.AUDIT_GENERATION: The TOE will record important user actions.
- O.AUDIT_REVIEW: The TOE will provide means for all recorded actions to be available for review by an System Security Officer.
- O.USER_IDENTIFICATION: The TOE will uniquely identify all users.
- OE.TIME: The TOE will ensure that all recorded actions have reliable timestamps.

8.1.1.2 P.AUTHORIZATION

The TOE shall limit the extent of each user's abilities in accordance with the TSP.

This Organizational Policy is satisfied by ensuring that:

- O.ACCESS: The TOE will ensure that access control decisions are enforced based on the applicable user and data security attributes and that administrators can manage user attributes.
- O.PROTECT: The TOE will ensure that access control decisions are enforced based on the applicable user and data security attributes and that users can manage access to their own data.
- O.USER_IDENTIFICATION: The TOE will uniquely identify each user.

8.1.1.3 P.AUTHORIZED_USERS

Access controls will ensure that only those users who have been authorized to access the protected information within the TOE will be able to do so.

This Organizational Policy is satisfied by ensuring that:

- O.ACCESS: The TOE will provide mechanisms to allow only authorized users to access the TOE, mainly Discretionary Access Controls.

8.1.1.4 P.I_AND_A

All users must be identified and authenticated prior to accessing any controlled resources with the exception of public objects.

This Organizational Policy is satisfied by ensuring that:

- O.USER_AUTHENTICATION: The TOE requires users to authenticate their identity prior to accessing any other functions.
- O.USER_IDENTIFICATION: The TOE requires users to claim their unique identity prior to accessing any other functions.

8.1.1.5 P.NEED_TO_KNOW

The TOE must limit the access to information in protected resources to those authorized users who have a need to know that information.

This Organizational Policy is satisfied by ensuring that:

- O.ACCESS: The TOE provides the administration functions to change a user's security attributes when that user no longer needs to access certain information.
- O.DISCRETIONARY_ACCESS: The TOE requires the resources to be protected according to the rules of the discretionary access control policy.
- O.PROTECT: The TOE requires the protection of resources.
- O.USER_IDENTIFICATION: The TOE requires access decision to be based on unique user identities.

8.1.1.6 P.ROLES

The TOE shall provide a system administrator and a system security officer role for secure administration of the TOE. This role shall be separate and distinct from other authorized users.

This Organizational Policy is satisfied by ensuring that:

- O.ADMIN_ROLE: The TOE has the objective of providing a system administrator and a system security officer role for secure administration. The TOE may provide other roles as well, but only the roles of system administrator and system security officer are required.

8.1.1.7 T.ADMIN_ERROR

A system administrator or a system security officer may incorrectly install or configure the TOE resulting in ineffective security mechanisms.

This Threat is satisfied by ensuring that:

- O.MANAGE: The TOE provides the administrator the necessary security management functions.

8.1.1.8 T.AUDIT_COMPROMISE

A process or user may cause audit data to be inappropriately accessed (viewed, modified or deleted), or prevent future records from being recorded, thus masking an attacker's actions.

This Threat is satisfied by ensuring that:

- O.AUDIT_GENERATION: The TOE will generate an audit log.
- O.AUDIT_PROTECTION: The TOE must also provide protection for its audit data.
- OE.PHYSICAL: The environment must address the possible compromise of audit data due to physical means.

8.1.1.9 T.MASQUERADE

An unauthorized user, process, or external IT entity may masquerade as an authorized entity to gain access to data or TOE resources.

This Threat is satisfied by ensuring that:

- O.USER_AUTHENTICATION: The TOE requires all users of the TOE to prove their claimed unique identity.
- O.USER_IDENTIFICATION: The TOE uniquely identifies each user.

8.1.1.10 T.RAW_ACCESS

An unauthorized user may gain access to sensitive information after obtaining access to the raw database storage medium.

This Threat is satisfied by ensuring that:

- O.CRYPTO: The TOE protects access to sensitive raw database information by offering the ability to encrypt sensitive data.

8.1.1.11 T.RESIDUAL_DATA

A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.

This Threat is satisfied by ensuring that:

- O.RESIDUAL_INFORMATION: The TOE prohibits users from accessing data that had been stored in system resources previously allocated to other users.

8.1.1.12 T.RESOURCE

An authenticated database user might consume excessive global database resources in a way which compromises the ability of other database users to access the DBMS.

This Threat is satisfied by ensuring that:

- O.RESOURCE: The TOE has the means of limiting the consumption of such resources, including the enforcement of limits on the number of concurrent sessions an individual may have.

8.1.1.13 T.SYSACC

A malicious process or user may gain unauthorized access to the system administrator or the system security officer account, or that of other trusted personnel.

This Threat is satisfied by ensuring that:

- O.ACCESS: The TOE prevents the wrong individuals from gaining unauthorized access to the system administrator's or the system security officer's account.
- O.MANAGE: The TOE provides mechanisms for the system administrator and the system security officer to set the security attributes for users so they are not allowed admin access.
- O.USER_AUTHENTICATION: The TOE requires the system administrator and the system security officer to be authenticated.
- O.USER_IDENTIFICATION: The TOE requires the system administrator and the system security officer to be uniquely identified.
- OE.PHYSICAL: The environment must address the possible unauthorized access to administrative accounts due to physical means.

8.1.1.14 T.TSF_COMPROMISE

A malicious user or process may cause TOE data to be inappropriately accessed (viewed, modified or deleted).

This Threat is satisfied by ensuring that:

- O.TOE_PROTECTION: The TOE protects TSF data and executable code.
- OE.TOE_PROTECTION: The IT environment protects its own data and executable code as well as that of its hosted applications.
- OE.PHYSICAL: The environment must protect the TSF data and executable code from a compromise through physical means.

8.1.1.15 T.UNAUTH_ACCESS

A user may gain unauthorized access (view, modify, delete) to user data.

This Threat is satisfied by ensuring that:

- O.ACCESS: The TOE ensures that only authorized users may gain access to the TOE and the resources it protects, and that users are not allowed to access protected data for which they are not authorized.
- O.DISCRETIONARY_ACCESS: The TOE controls access to user data by a discretionary access control policy.
- O.INTERNAL_TOE_DOMAINS: The TOE maintains internal domains to keep data and processes of concurrent users separate, so users cannot observe or interfere with other users' data or queries.
- O.PROTECT: The TOE prevents unauthorized access to user data.
- OE.PHYSICAL: The environment must prevent unauthorized physical access to the TOE.

8.1.1.16 T.UNDETECTED_ACTIONS

Failure of the IT operating system to detect and record attempts to perform unauthorized actions may occur.

This Threat is satisfied by ensuring that:

- O.AUDIT_GENERATION: The TOE detects and records security relevant actions.
- O.AUDIT_PROTECTION: The TOE prevents unauthorized modification. Of audit records
- OE.TIME: The TOE includes reliable timestamps with each audit record.
- OE.PHYSICAL: The environment must prevent potentially undetected physical manipulation of the TOE.

8.1.1.17 T.UNIDENTIFIED_ACTIONS

Failure of the system administrator and the system security officer to identify and act upon attempts to perform unauthorized actions may occur.

This Threat is satisfied by ensuring that:

- O.AUDIT_REVIEW: The TOE provides the tools to effectively review audit records.
- O.MANAGE: The TOE provides necessary access to the audit trail.

8.1.1.18 A.NO_EVIL

System administrators and the system security officers are non-hostile, appropriately trained and follow all administrator guidance.

This Assumption is satisfied by ensuring that:

- OE.CONFIG: System administrators and the system security officers are trained and trusted to properly configure the IT environment so it enforces its security policies.

8.1.1.19 A.NO_GENERAL_PURPOSE

There are no general-purpose computing capabilities (e.g., compilers or user applications) available on DBMS servers, other than those services necessary for the operation, administration and support of the DBMS.

This Assumption is satisfied by ensuring that:

- OE.NO_GENERAL_PURPOSE: The DBMS server must not include any general-purpose commuting or storage capabilities. This will protect the TSF data from malicious processes.

8.1.1.20 A.PHYSICAL

It is assumed that appropriate physical security is provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.

This Assumption is satisfied by ensuring that:

- OE.PHYSICAL: The TOE, the TSF data, and protected user data is assumed to be protected from physical attack (e.g., theft, modification, destruction, or eavesdropping). Physical attack could include unauthorized intruders into the TOE environment, but it does not include physical destructive actions that might be taken by an individual that is authorized to access the TOE environment.

8.1.1.21 A.ROBUST_ENVIRONMENT

It is assumed that the IT environment provides support commensurate with the expectations of the TOE.

This Assumption is satisfied by ensuring that:

- OE.TRUST_IT: The IT entities in the environment are correctly installed, configured, managed, maintained and provide the applicable security functions.

8.1.1.22 A.NETWORK

It is assumed that the environment protects network communication media appropriately.

This Assumption is satisfied by ensuring that:

- OE.NETWORK: The environment is responsible to protect network traffic to and from the TOE from unauthorized disclosure. Note that this is an environment objective since the possible mechanisms could range from physical protection of the network media to cryptographic tunneling.

8.2 Security Requirements Rationale

This section provides evidence supporting the internal consistency and completeness of the components (requirements) in the Security Target. Note that **Table 6** indicates the requirements that effectively satisfy the individual objectives. .

8.2.1 Security Functional Requirements Rationale

All Security Functional Requirements (SFR) identified in this Security Target are fully addressed in this section and each SFR is mapped to the objective for which it is intended to satisfy.

	O.ACCESS	O.ADMIN_ROLE	O.AUDIT_GENERATION	O.AUDIT_PROTECTION	O.AUDIT_REVIEW	O.CRYPTO	O.DISCRETIONARY_ACCESS	O.INTERNAL_TOE_DOMAINS	O.MANAGE	O.PROTECT	O.RESIDUAL_INFORMATION	O.RESOURCE	O.TOE_PROTECTION	O.USER_AUTHENTICATION	O.USER_IDENTIFICATION	OE.TIME	OE.TOE_PROTECTION
FAU_GEN.1			X														
FAU_GEN.2			X														
FAU_SAR.1					X												
FAU_SAR.2				X													
FAU_SAR.3					X												
FAU_SEL.1			X														
FAU_STG.1				X													
FAU_STG.3				X													

	O.ACCESS	O.ADMIN_ROLE	O.AUDIT_GENERATION	O.AUDIT_PROTECTION	O.AUDIT_REVIEW	O.CRYPTO	O.DISCRETIONARY_ACCESS	O.INTERNAL_TOE_DOMAINS	O.MANAGE	O.PROTECT	O.RESIDUAL_INFORMATION	O.RESOURCE	O.TOE_PROTECTION	O.USER_AUTHENTICATION	O.USER_IDENTIFICATION	OE.TIME	OE.TOE_PROTECTION
FCS_CKM.1						X											
FCS_CKM.4						X											
FCS_COP.1a						X											
FCS_COP.1b						X											
FDP_ACC.1a	X						X			X							
FDP_ACC.1b	X									X							
FDP_ACF.1a	X						X			X							
FDP_ACF.1b	X									X							
FDP_RIP.2a										X	X						
FIA_AFL.1														X			
FIA_ATD.1															X		
FIA_SOS.1														X			
FIA_UAU.2														X			
FIA_UAU.7														X			
FIA_UID.2															X		
FIA_USB.1			X				X								X		
FMT_MOF.1			X			X			X								
FMT_MSA.1							X		X								
FMT_MSA.2									X								
FMT_MSA.3							X										
FMT_MTD.1a									X								
FMT_MTD.1b				X					X								
FMT_MTD.1c									X					X			
FMT_MTD.1d						X											
FMT_REV.1a	X																
FMT_REV.1b										X							
FMT_SMF.1			X			X	X		X								
FMT_SMR.1		X															
FRU_RSA.1												X					
FPT_RVM.1a								X					X				
FPT_SEP.1a								X					X				
FTA_MCS_EXP.1												X					
FTA_TSE.1	X																
FDP_ACC.1c																	X
FDP_ACF.1c																	X
FDP_RIP.2b																	X
FPT_RVM.1b																	X
FPT_SEP.1b																	X
FPT_STM.1			X													X	

Table 6 Objective to Requirement Correspondence

8.2.1.1 O.ACCESS

The TOE will ensure that users gain only authorized access to it and to the resources that it controls.

This TOE Security Objective is satisfied by ensuring that:

- FDP_ACC.1a: The Discretionary Access Control policy applies to all operations between subjects and objects (databases, tables, views, and stored procedures) controlled by the TOE.
- FDP_ACC.1b: The Policy-based Access Control policy applies to select, update, and delete operations between subjects and objects (rows) controlled by the TOE.
- FDP_ACF.1a: The subjects and objects under the discretionary access control policy will have certain rules that apply to all accesses between them. The rules are based on subject identities and access control lists associated with database objects that either grant or deny potential request access.
- FDP_ACF.1b: The subjects and objects under the policy-based access control policy will have certain rules that apply to all accesses between them. The rules are based on application contexts determined during session establishment that can be assigned to database columns to control access to individual rows in a table.
- FMT_REV.1a: Security attributes associated with subjects and objects are the basis for access control. Revocation of these security attributes would modify the access control policy. The system administrator and the system security officer should have control over security attributes associated with users (such as user authentication data), being the only role that can revoke them.
- FTA_TSE.1: The TOE can restrict access to itself (i.e., session establishment) based on the time.

8.2.1.2 O.ADMIN_ROLE

The TOE will provide system administrator and system security officer roles to isolate administrative actions.

This TOE Security Objective is satisfied by ensuring that:

- FMT_SMR.1: The TOE will establish, at least, a system administrator and a system security officer role. The system administrator and system security officer will be given privileges to perform certain tasks that other users will not be able to perform. These privileges include, but are not limited to, access to audit information and security functions.

8.2.1.3 O.AUDIT_GENERATION

The TOE will provide the capability to detect and create records of security relevant events associated with users.

This TOE Security Objective is satisfied by ensuring that:

- FAU_GEN.1: This objective is satisfied in part by the requirement that the TOE generate audit records according to the minimum level of auditing, as defined by the Common Criteria.
- FAU_GEN.2: Each audit record written must be descriptive of the event that caused a record to be generated, and must be associated with the unique identity of the user that caused the event.
- FAU_SEL.1: The TOE enables the system security officer to pre-select events to include in the audit log.
- FIA_USB.1: All subjects that act on behalf of users must have a binding that associates the subjects with a user. This is necessary to be able to associate audit records with user identities.
- FMT_MOF.1: The TOE ensures that the system security officer role is the only role authorized to manipulate the behavior of the audit generation mechanism.
- FMT_SMF.1: The TOE ensures that the system security officer role is able to manipulate the behavior of the audit generation mechanism.
- FPT_STM.1: Reliable time stamps are assumed to be provided by the IT environment.

8.2.1.4 O.AUDIT_PROTECTION

The TOE will provide the capability to protect audit information.

This TOE Security Objective is satisfied by ensuring that:

- FAU_SAR.2: Users must not be able to read the audit records, unless they have been granted explicit readaccess to the audit log.
- FAU_STG.1: The TOE prevents unauthorized deletion or modification of audit records.
- FAU_STG.3: The TOE provides site-configurable options to prevent loss of audit data in the event the audit storage space is exhausted.
- FMT_MTD.1b: Only the system security officer has the ability to query or clear audit records.

8.2.1.5 O.AUDIT_REVIEW

The TOE will provide the capability to selectively view audit information.

This TOE Security Objective is satisfied by ensuring that:

- FAU_SAR.1: In order for the system security officer to review the audit logs they must be accessible in a suitable form for the system security officer to read, which means the system security officer should have the appropriate functions needed to interpret the data.
- FAU_SAR.3: The system security officer must be able to search and sort on the audit data based on date, time, type of event, event status (success or failure), or user identity. This will allow the system security officer to examine specific events more efficiently.

8.2.1.6 O.CRYPTO

The TOE will provide the capability to encrypt sensitive data.

This TOE Security Objective is satisfied by ensuring that:

- FCS_CKM.1: The TOE creates appropriate cryptographic keys for use in the encryption and decryption operations.
- FCS_CKM.4: The TOE destroys cryptographic keys when they are no longer needed by the TOE.
- FCS_COP.1a: The TOE automatically encrypts and decrypts the contents of columns that have been configured to be encrypted.
- FCS_COP.1b: The TOE uses SHA-1 hashing to support the generation of keys in accordance with KDF1.
- FMT_MOF.1: Only system security officers may enable and disable the cryptographic functions of the TOE.
- FMT_MTD.1d: Only system security officers can configure critical cryptographic key related data within the TOE.
- FMT_SMF.1: System security officers are enabled to manage the cryptographic function.

8.2.1.7 O.DISCRETIONARY_ACCESS

The TOE will control access to resources based upon the identity of users ,active roles in a user session, group membership of users, object ownership and access control lists.

This TOE Security Objective is satisfied by ensuring that:

- FDP_ACC.1a: The Discretionary Access Control policy applies to all operations between subjects and objects controlled by the TOE.
- FDP_ACF.1a: The subjects and objects under the discretionary access control policy will have certain rules that apply to all accesses between them. The rules are based on subject identities and access control lists associated with database objects that either grant or deny potential request access.
- FIA_USB.1: All subjects that act on behalf of users must have a binding that associates the subjects with a user uniquely.
- FMT_MSA.1: Only system administrators, system security officers and database owners may manipulate the security attributes of database users.

- FMT_MSA.3: Default access control attributes are restrictive to prevent accidental (non-discretionary) disclosure of information that should be protected.
- FMT_SMF.1: System administrators, system security officers and database owners must be able to manipulate the security attributes of database users.

8.2.1.8 O.INTERNAL_TOE_DOMAINS

The TSF will maintain internal domains for separation of data and queries belonging to concurrent users.

This TOE Security Objective is satisfied by ensuring that:

- FPT_RVM.1a: The mechanisms providing self-protection are always invoked and not able to be bypassed.
- FPT_SEP.1a: The TSF enforces separation between the security domains within its scope of control.

8.2.1.9 O.MANAGE

The TOE will provide all the functions and facilities necessary to support the system administrators, system security officers and database owners in their management of the security of the TOE.

This TOE Security Objective is satisfied by ensuring that:

- FMT_MOF.1: Only the system security officer will be able to enable or disable functions of the audit log. This will prevent a malicious user from turning off the audit log while he/she performs a malicious act, then turning it back on when he/she is done.
- FMT_MSA.1: Only system administrators, system security officers and database owners may manipulate the security attributes of database users.
- FMT_MSA.2: The TOE rejects invalid and insecure data to help ensure the effectiveness of the security functions.
- FMT_MTD.1a: Only system security officers are able to manage the inclusion/exclusion of specific events to be audited.
- FMT_MTD.1b: Only system security officers are authorized to query or clear the audit log.
- FMT_MTD.1c: Only system security officers or the users themselves are authorized to set or reset user authentication data.
- FMT_SMF.1: The system security officer will be able to enable or disable functions of the audit log, select audited events, review audit records, and manage database subjects and authentication data.

8.2.1.10 O.PROTECT

The TOE will provide mechanisms to protect user data and resources.

This TOE Security Objective is satisfied by ensuring that:

- FDP_ACC.1a: The Discretionary Access Control policy applies to all operations between subjects and objects (databases, tables, views, and stored procedures) controlled by the TOE.
- FDP_ACC.1b: The Policy-based Access Control policy applies to select, update, and delete operations between subjects and objects (rows) controlled by the TOE.
- FDP_ACF.1a: The subjects and objects under the discretionary access control policy will have certain rules that apply to all accesses between them. The rules are based on subject identities and access control lists associated with database objects that either grant or deny potential requested access.
- FDP_ACF.1b: The subjects and objects under the policy-based access control policy will have certain rules that apply to all accesses between them. The rules are based on application contexts determined during session establishment that can be assigned to database columns to control access to individual rows in a table.
- FDP_RIP.2a: When data is deleted from memory or storage (e.g., disk drive) it is often left intact and not truly erased. It is then possible for other users with access to the memory to view previously protected data. Therefore when a block of memory is allocated it is necessary to ensure all previous data stored in that block has been made unavailable.
- FMT_REV.1b:. The discretionary nature of the policy allows users to modify access control permissions, which are represented by security attributes. Users are allowed to modify the security attributes of subjects and objects as permitted by the Discretionary Access Control policy.

8.2.1.11 O.RESIDUAL_INFORMATION

The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.

This TOE Security Objective is satisfied by ensuring that:

- FDP_RIP.2a: When data is deleted from memory or storage (e.g., disk drive) it is often left intact and not truly erased. It is then possible for other users with access to the memory to view previously protected data. Therefore when a block of memory is allocated it is necessary to ensure all previous data stored in that block has been made unavailable.

8.2.1.12 O.RESOURCE

The TOE must provide the means of controlling the consumption of database resources by authorised users of the TOE.

This TOE Security Objective is satisfied by ensuring that:

- FRU_RSA.1: The TOE is required to limit database queries and transactions so that they do not exceed specified I/O costs limits, return too many rows, and do not exceed a specified elapsed time limit.
- FTA_MCS_EXP.1: The TOE is required to allow administrator to restrict the maximum number of concurrent user sessions.

8.2.1.13 O.TOE_PROTECTION

The TOE will protect itself and its assets from external interference or tampering.

This TOE Security Objective is satisfied by ensuring that:

- FPT_RVM.1a: The TOE is required to allow access to protected objects only after it makes informed access decisions.
- FPT_SEP.1a: The TOE is required to protect itself and separate the contexts of its users.

8.2.1.14 O.USER_AUTHENTICATION

The TOE will verify the claimed identity of users.

This TOE Security Objective is satisfied by ensuring that:

- FIA_AFL.1: To prevent brute force attacks on authentication data, the administrator must specify an upper bound on the number of unsuccessful authentications that will be allowed. Surpassing that threshold could indicate a brute force user authentication attack, and the TOE needs to take appropriate action.
- FIA_SOS.1: User authentication is meaningful only if there is an extremely low probability of success for random attempts to authenticate as an authorized user. The requirement ensures that the secret authentication data is computationally difficult to guess randomly.
- FIA_UAU.2: Users must be authenticated before they can perform any TSF-mediated functions.
- FIA_UAU.7: To mitigate the chance of one user masquerading as another, the TOE will not echo passwords and will not provide information that specifically identifies the nature of applicable authentication failures.
- FMT_MTD.1c: The user authentication data is to be set only by an authenticated individual in an authorized role.

8.2.1.15 O.USER_IDENTIFICATION

The TOE will uniquely identify users.

This TOE Security Objective is satisfied by ensuring that:

- FIA_ATD.1: Each database user will have a list of security attributes associated with them. They will have their unique identifier, any groups they may be a part of, for discretionary access control, any security roles they posses, and any other attributes assigned by the ST writer.
- FIA_UID.2: Users must be identified to the TOE before they can perform any TSF-mediated functions.

- FIA_USB.1: All subjects that act on behalf of users must have a binding that associates the subjects with a user uniquely.

8.2.1.16 OE.TIME

The IT environment will provide a time source that provides reliable time stamps.

This IT Environment Security Objective is satisfied by ensuring that:

- FPT_STM.1: The IT environment is required to provide a reliable time source.

8.2.1.17 OE.TOE_PROTECTION

The IT environment will provide protection to the TOE and its assets from external interference, tampering, and disclosure.

This IT Environment Security Objective is satisfied by ensuring that:

- FDP_ACC.1c: The Environment Access Control policy applies to access operations between processes and files and shared memory controlled by the IT environment.
- FDP_ACF.1c: The subjects and objects under the environment access control policy have rules that apply to all accesses between them. The rules are based on process identities and access permissions associated with files and shared memory that either grant or deny potential requested access.
- FDP_RIP.2b: The IT environment is responsible to ensure that it does not provide potential residual information when providing object constructs to its hosted applications.F
- FPT_RVM.1b: The IT environment is required to allow access to protected objects only after it makes informed access decisions.
- FPT_SEP.1b: The IT environment is required to protect itself and separate the contexts of its users.

8.3 Security Assurance Requirements Rationale

Adaptive Server Enterprise is targeted at a generalized IT environment with good physical access security and competent administrators. Within such environments it is assumed that attackers will have a low to moderate attack potential. As such, EAL 4 (augmented with ALC_FLR.2) is appropriate to provide the assurance necessary to counter the potential for attack. Note also that this security target has defined an environment requiring more security than the U.S. Government Protection Profile Consistency Guidance for Basic Robustness, dated 24 July 2002, and that is comparable to or better than the historical notion of the B1 level of the Trusted Computer System Evaluation Criteria.

8.4 Strength of Function Rationale

Adaptive Server Enterprise is targeted at a generalized IT environment with good physical access security and competent administrators. Within such environments it is assumed that attackers will have a moderate attack potential. As such, a strength of functions claim of ‘medium’ is appropriate for the intended environment. Note that the only applicable mechanisms (i.e., those that are probabilistic or permutational) are related to identification and authentication (FIA_SOS.1, FIA_UAU.2, and FIA_UID.2).

8.5 Requirement Dependency Rationale

The following table represents an analysis of the dependencies of the security functional requirements (SFRs) in this security target. The first column identifies all of the SFRs in this security target. The TOE SFRs are highlighted in **bold**, unlike the IT environment SFRs. The second column identifies the minimum dependencies defined in the Common Criteria v2.3. The third column identifies the actual requirements in this security target that correspond to the identified dependencies. Again, the corresponding TOE SFRs are highlighted in **bold**. Notice that this table demonstrates that all of the identified dependencies are satisfied. It also shows that the TOE has some dependencies on the IT environment, but the requirements for the IT environment have been defined such that it is not dependent upon the TOE.

With regard to the security assurance requirements (SARs), EAL 4 has been adopted from the Common Criteria and it is assumed that it has been designed so that it fulfills the applicable dependencies. The only additional SAR in this security target is ALC_FLR.2, which has no dependencies.

ST Requirement	CC Dependencies	ST Dependencies
FAU_GEN.1	FPT_STM.1	FPT_STM.1
FAU_GEN.2	FAU_GEN.1 FIA_UID.1	**FAU_GEN.1** **FIA_UID.2**
FAU_SAR.1	FAU_GEN.1	**FAU_GEN.1**
FAU_SAR.2	FAU_SAR.1	**FAU_SAR.1**
FAU_SAR.3	FAU_SAR.1	**FAU_SAR.1**
FAU_SEL.1	FAU_GEN.1 and FMT_MTD.1	**FAU_GEN.1** **FMT_MTD.1a**
FAU_STG.1	FAU_GEN.1	**FAU_GEN.1**
FAU_STG.3	FAU_STG.1	**FAU_STG.1**
FCS_CKM.1	(FCS_CKM.2 or FCS_COP.1) and FCS_CKM.4 and FMT_MSA.2	**FCS_COP.1** **FCS_CKM.4** **FMT_MSA.2**
FCS_CKM.4	(FDP_ITC.1, FDP_ITC.2 or FCS_CKM.1) and FMT_MSA.2	**FCS_CKM.1** **FMT_MSA.2**
FCS_COP.1a	(FDP_ITC.1, FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4 and FMT_MSA.2	**FCS_CKM.1** **FCS_CKM.4** **FMT_MSA.2**
FCS_COP.1b	(FDP_ITC.1, FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4 and FMT_MSA.2	**FCS_CKM.1** **FCS_CKM.4** **FMT_MSA.2**
FDP_ACC.1a	FDP_ACF.1	**FDP_ACF.1a**
FDP_ACC.1b	FDP_ACF.1	**FDP_ACF.1b**
FDP_ACF.1a	FDP_ACC.1	**FDP_ACC.1a**
FDP_ACF.1b	FDP_ACC.1	**FDP_ACC.1b**
FDP_RIP.2a	none	none
FIA_AFL.1	FIA_UAU.1	**FIA_UAU.2**
FIA_ATD.1	none	none
FIA_SOS.1	none	none
FIA_UAU.2	FIA_UID.1	**FIA_UID.2**
FIA_UAU.7	FIA_UAU.1	**FIA_UAU.2**
FIA_UID.2	none	none
FIA_USB.1	FIA_ATD.1	**FIA_ATD.1**
FMT_MOF.1	FMT_SMR.1 and FMT_SMF.1	**FMT_SMR.1** **FMT_SMF.1**
FMT_MSA.1	FMT_SMR.1 and FMT_SMF.1 and (FDP_ACC.1 or FDP_IFC.1)	**FMT_SMR.1** **FMT_SMF.1** **FDP_ACC.1**
FMT_MSA.2	ADV_SPM.1 and FMT_MSA.1 and FMT_SMR.1 and (FDP_ACC.1 or FDP_IFC.1)	**ADV_SPM.1** **FMT_MSA.1** **FMT_SMF.1** **FDP_ACC.1**
FMT_MSA.3	FMT_MSA.1 and FMT_SMR.1	**FMT_MSA.1** **FMT_SMR.1**
FMT_MTD.1a	FMT_SMR.1 and FMT_SMF.1	**FMT_SMR.1** **FMT_SMF.1**
FMT_MTD.1b	FMT_SMR.1 and	**FMT_SMR.1**

ST Requirement	CC Dependencies	ST Dependencies
	FMT_SMF.1	**FMT_SMF.1**
FMT_MTD.1c	FMT_SMR.1 and FMT_SMF.1	**FMT_SMR.1** **FMT_SMF.1**
FMT_MTD.1d	FMT_SMR.1 and FMT_SMF.1	**FMT_SMR.1** **FMT_SMF.1**
FMT_REV.1a	FMT_SMR.1	**FMT_SMR.1**
FMT_REV.1b	FMT_SMR.1	**FMT_SMR.1**
FMT_SMF.1	none	none
FMT_SMR.1	FIA_UID.1	**FIA_UID.2**
FPT_RVM.1a	none	none
FPT_SEP.1a	none	none
FRU_RSA.1	none	none
FTA_MCS_EXP.1	FIA_UID.1	**FIA_UID.2**
FTA_TSE.1	none	none
FDP_ACC.1c	FDP_ACF.1	FDP_ACF.1c
FDP_ACF.1c	FDP_ACC.1	FDP_ACC.1c
FDP_RIP.2b	none	none
FPT_RVM.1b	none	none
FPT_SEP.1b	none	none
FPT_STM.1	none	none

Table 7 Requirement Dependencies

8.6 Explicitly Stated Requirements Rationale

This security target includes one explicitly stated requirement: FTA_MCS_EXP.1. This requirement is very similar to the CC Part 2 FTA_MCS.1 requirement; except that it only requires that the TOE *must be able* to limit concurrent user sessions as opposed to requiring that it always *must* do so. This explicit requirement was necessary since the CC does not provide the flexibility of having an optionally configured mechanism. As such, FTA_MCS_EXP.1 should be considered as an alternate version of FTA_MCS.1 that shares the same requirement class and family as well as dependencies.

8.7 TOE Summary Specification Rationale

Each subsection in Section 6, the TOE Summary Specification, describes a security function of the TOE. Each description is followed with rationale that indicates which requirements are satisfied by aspects of the corresponding security function. The set of security functions work together to satisfy all of the security functions and assurance requirements. Furthermore, all of the security functions are necessary in order for the TSF to provide the required security functionality.

This Section in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions are suitable to meet the TOE security requirements. The collection of security functions work together to provide all of the security requirements. The security functions described in the TOE summary specification are all necessary for the required security functionality in the TSF. **Table 8 Security Functions vs. Requirements Mapping** demonstrates the relationship between security requirements and security functions.

	Security audit	Cryptographic support	User data protection	Identification and authentication	Security management	Protection of the TSF	Resource Utilization	TOE access
FAU_GEN.1	X							
FAU_GEN.2	X							
FAU_SAR.1	X							
FAU_SAR.2	X							
FAU_SAR.3	X							
FAU_SEL.1	X							
FAU_STG.1	X							
FAU_STG.3	X							
FCS_CKM.1		X						
FCS_CKM.4		X						
FCS_COP.1a		X						
FCS_COP.1b		X						
FDP_ACC.1a			X					
FDP_ACC.1b			X					
FDP_ACF.1a			X					
FDP_ACF.1b			X					
FDP_RIP.2a			X					
FIA_AFL.1				X				
FIA_ATD.1				X				
FIA_SOS.1				X				
FIA_UAU.2				X				
FIA_UAU.7				X				
FIA_UID.2				X				
FIA_USB.1				X				
FMT_MOF.1					X			
FMT_MSA.1					X			
FMT_MSA.2					X			
FMT_MSA.3					X			
FMT_MTD.1a					X			
FMT_MTD.1b					X			
FMT_MTD.1c					X			
FMT_MTD.1d					X			
FMT_REV.1a					X			
FMT_REV.1b					X			
FMT_SMF.1					X			
FMT_SMR.1					X			
FPT_RVM.1a						X		
FPT_SEP.1a						X		
FRU_RSA.1							X	
FTA_MCS_EXP.1								X
FTA_TSE.1								X

Table 8 Security Functions vs. Requirements Mapping

8.8 PP Claims Rationale

See Section 7, Protection Profile Claims.