IBM® DB2® Content Manager Enterprise Edition V8.4 Fix Pack 1a Security Target

Version 1.0

22 December 2008

Prepared For:
International Business Machines (IBM)
555 Bailey Avenue
San Jose, CA 95161

Prepared By:
Science Applications International Corporation
Common Criteria Testing Laboratory
7125 Gateway Drive
Columbia, MD 21046

Table of Contents

List of Figures and Tables

1 Security Target Introduction

This section provides the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is the IBM® DB2® Content Manager Enterprise Edition Version 8.4 Fix Pack 1a product, provided by International Business Machines (IBM). IBM® DB2® Content Manager Enterprise Edition is a data management system (content management system) that provides a foundation for managing, accessing, and integrating critical business information on demand.

The Security Target contains the following additional sections:

- Target of Evaluation (TOE) Description—provides an overview of the TOE, describes the TOE in terms of its physical and logical boundaries, and states the scope of the TOE
- TOE Security Environment—identifies and describes organizational security policies to be met by the TOE and assumptions about the intended environment and method of use of the TOE
- Security Objectives—identifies and describes the security objectives for the TOE and its environment
- IT Security Requirements—presents the security functional requirements (SFRs) for the TOE and the IT Environment that supports the TOE, and the security assurance requirements (SARs) against which the TOE is evaluated
- TOE Summary Specification—describes the TOE security functions and the assurance measures that satisfy the security requirements.
- Protection Profile Claims—identifies any Protection Profile claims made in the ST.
- Rationale—documents the justifications of the security objectives, security requirements and TOE summary specification as to their consistency, completeness and suitability.

1.1 Security Target, TOE, and CC Identification

ST Title – IBM® DB2® Content Manager Enterprise Edition V8.4 Fix Pack 1a Security Target

ST Version – 1.0

ST Date – 22 December 2008

TOE Identification – IBM® DB2® Content Manager Enterprise Edition V8.4 Fix Pack 1a

(see section 2.4.2 for a list of supported operating systems and other product dependencies and section 2.4.3 for a list of products included with the Content Manager distribution that are excluded from the TOE)

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

1.2 Common Criteria Conformance Claims

This TOE and ST are consistent with the following specifications:

- Common Criteria (CC) for Information Technology (IT) Security Evaluation Part 2: Security functional requirements, Version 2.3, August 2005.
    - Part 2 extended
- Common Criteria (CC) for Information Technology Security Evaluation Part 3: Security assurance requirements, Version 2.3, August 2005.
    - Part 3 conformant
    - Evaluation Assurance Level 4 (EAL4) augmented with ALC_FLR.2
- The ST claims a minimum strength of function of SOF-Medium for the TOE.

1.3 Conventions, Terminology, and Acronyms

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 FCS_COP.1(a) and FCS_COP.1(b) indicate that the ST includes two iterations of the FCS_COP.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 "…**big** ~~some~~ things …").
- Explicitly stated requirements are identified with 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 Terminology

The terminology used in this Security Target is defined below:

Administrative domain A section of a Library Server that one or more administrators manage. An administrative domain limits administrative and user access to a section of the Library Server.

Authorized users The users, administrative and non-administrative, who have been given access to the TOE.

Collection A group of objects with a similar set of management rules.

Connectors Object-oriented programming class that provides standard access to APIs native to specific content servers.

Event log An audit record in the event tables.

Item In DB2 Content Manager, generic term for an instance of an *item type*. For example, an item might be a folder, document, video, or image.

Item type A template for defining and later locating like *items*, consisting of a root component, zero or more child components, and a categorization.

Privilege A privilege is the right to act on an object in a specific way.

User exit A point in the execution of the TOE at which a user exit routine can be given control.

User exit routine A user routine that receives control at predefined user exits. It could be written by the user, but default user exit routines are also provided as part of the TOE.

User Group A group of individual users who perform similar tasks.

Resource Any data entity that is stored on a resource manager in digital form. Objects can include, but are not limited to, JPEG images, MP3 audio, AVI video, a plain text file. For example, a few of the formats that are supported natively by Content Manager are: Microsoft Word, Lotus® WordPro, TIFF, and JPEG.

1.3.3 Abbreviations

The abbreviations used within this Security Target are expanded below:

ACL	Access Control List
AES	Advanced Encryption Standard
API	Application Programming Interface
CC	Common Criteria
CM	Configuration Management
EAL	Evaluation Assurance Level
FIPS	Federal Information Processing Standard
IBM	International Business Machines
ID	Identification
IT	Information Technology
NIST	National Institute of Standards and Technology
PC	Personal Computer
SAR	Security Assurance Requirement
SFP	Security Function Policy
SFR	Security Functional Requirement
ST	Security Target
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSP	TOE Security Policy
XML	Extensible Markup Language

2 Target of Evaluation (TOE) Description

The TOE is IBM® DB2® Content Manager Enterprise Edition V8.4 Fix Pack 1a, henceforth referred to as Content Manager.

2.1 Product Type

Content Manager is a data management system (content management system) that provides a foundation for managing, accessing, and integrating critical business information on demand. Content Manager is able to integrate all forms of data — document, Web, image, rich media — across diverse business processes and applications, including Siebel, PeopleSoft, and SAP, presenting the data in an integrated context for later use.

2.2 Product Description

The components of Content Manager comprise: a Library Server; one or more Resource Managers, the Content Manager 8.4 Connector Application Programming Interfaces (APIs); the System Administration Client; and the Client for Windows.

The Library Server is the key component of the Content Manager system. The Library Server resides on a DB2 Enterprise Server database environment. It is called the Library Server because it performs the functions that a library catalog file in a real library performs. The Library Server manages the content metadata and is responsible for identification and authentication for non-administrative users and identification for administrative users requesting services from Content Manager and access control to the resources residing on Resource Managers. The Library Server manages the relationships between items in the system and controls access to all of the system information, including the information stored in the Resource Managers. The Library Server processes requests (such as update or delete) from one or more clients. In Content Manager, all access to the Library Server is via stored procedures. The Library Server code is co-resident with the database engine code. The Library Server passes back to the client query results that include tokens and locators for requested content that the user is authorized to access. The database is not part of the TOE.

The Resource Manager stores resources for Content Manager. It can be on the same server as the Library Server, or it can be on its own computer. Resource Managers can be distributed across networks to provide convenient user access. Users store and retrieve digital resources on the Resource Manager by routing requests through the Library Server. A single Library Server can support multiple Resource Managers and content can be stored on any of these Resource Managers. When the Library Server grants an access request, the Library Server returns a token and the location of the object to the user. Data objects are always associated with a specific collection on a Resource Manager. Access decisions to grant access to a collection of data objects are made by the Library Server and enforced by the Resource Manager. The client communicates directly with the Resource Manager using Internet protocols. Tokens received from the Library Server are passed to Resource Managers from a client through the APIs to provide assurance that the request has been authorized and the access control information has not been altered since leaving the Library Server. The Resource Manager requires the following components in the IT environment (both of which are provided in the Content Manager installation package as a convenience to users):

- DB2 Enterprise Server database—required to run the Resource Manager database, which stores information pertaining to the objects being managed by the Resource Manager
- WebSphere Application Server—required to run the Resource Manager, which is implemented as a Java2 Enterprise Edition (J2EE) web application.

The Content Manager 8.4 Connector APIs (used by WebSphere Application Server applications, the System Adminsitration Client, and Clint for Windows) comprise a set of object-oriented APIs that allow applications and users to access the Library Server and Resource Manager(s) and are used to facilitate all functions within the TOE, including administrative functions. Note that these APIs are identified in the three boxes labeled 'APIs' in the figure below.

The System Administration Client oversees the entire Content Manager system. From the System Administration Client, an administrator performs various administrative functions, such as defining the data model, creating users and defining their access to the system and specific objects, and managing storage and storage objects in the system.

The System Administration Client can be installed on any workstation with the other components or on its own workstation. The Client for Windows provides an interface that enables an application to import documents into Content Manager, view them, work with them, store them, and retrieve them. Note that the System Administration Client and Client for Windows are both part of the TOE and serve to facilitate human access to their underlying programmatic APIs.

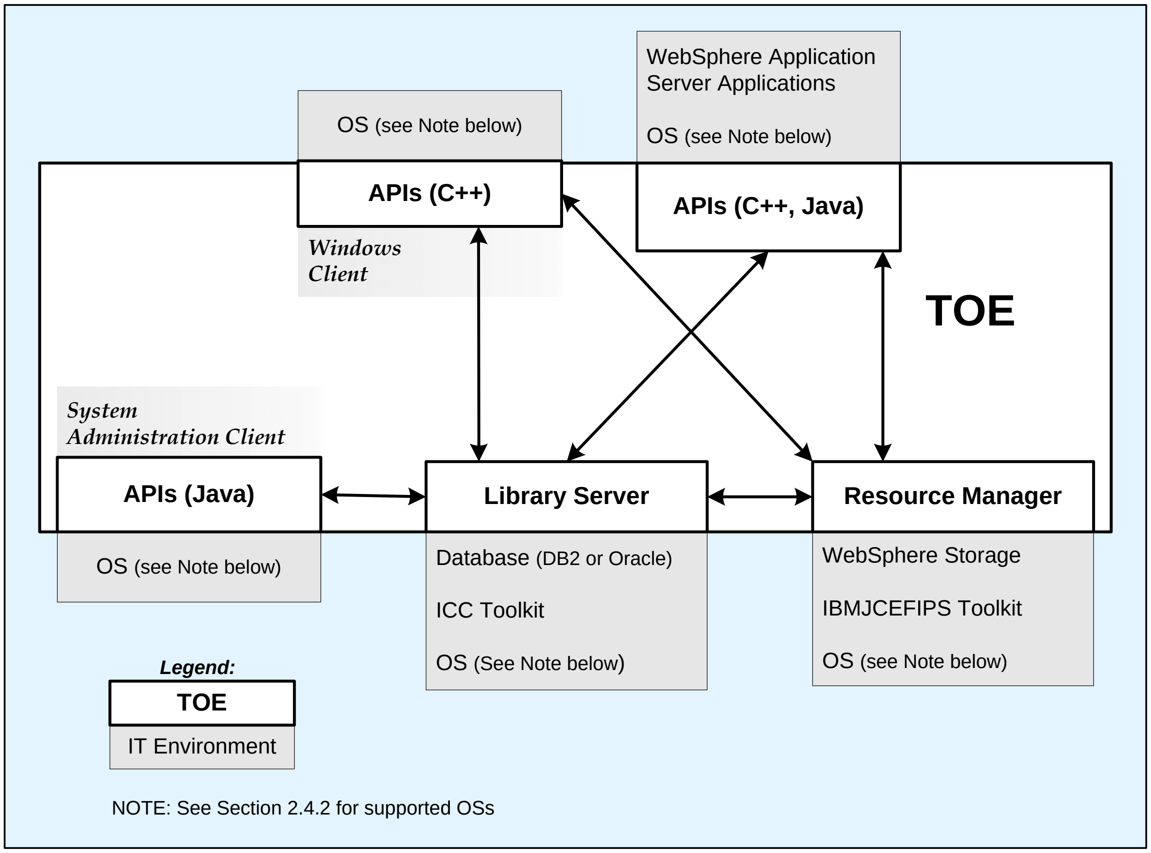

Figure 1: Content Manager Architecture

In Figure 1, the communication between the TOE components: Client for Windows, System Administration Client, Library Server, Resource Manager and the set of APIs should be protected as deemed necessary. This ST assumes that the channels would be protected to the degree necessary by available external means (e.g., physical network protection or some VPN technology).

2.3 Product Features

Embedded Database Engine

- All library server logic in Content Manager runs within DB2 Enterprise Server database. In effect, this architecture implements a data model within the relational database engine that is more appropriate for managing unstructured information than the relational model of tables, rows and columns. Stored procedures map the data model without executing logic in the client or a mid-tier application. Thus, applications built on this new model do not pay the performance penalty that an intermediate mapping layer requires. Equally important, the new data model inherits many key values and attributes of the mature relational system, like transactional and data integrity.

Advanced Data Modeling Capability

- Content Manager acts as the central authority for correlating diverse terms used for the same business attribute and for simplifying navigation and access to information for all authorized users and applications.
- Content Manager stores and manages indexing attributes in its library server, whereas objects are stored and managed in one or more associated resource managers. The following object attributes are managed:
  - Relationships to other objects
  - Access control, including who can access the object and the actions that authorized users can perform
  - Storage profile for hierarchical storage management
  - Lifecycle and retention
  - Workflow initiation, process integration and automation

Flexible Data Model

- The Content Manager data model supports hierarchical structures such as parent-child and peer-to-peer relationships.
- Attributes for an object can be structured with parent and child relationships that match the hierarchical structure in real-world customer application environments.
- It allows the creations of objects that combine attributes from different business processes and centralize information as needed.

Peer-to-peer Relationships: Links and References

- Content Manager allows custom applications to build more complex inter-object peer-to-peer relationships using links and references.

  Links have the following characteristics:

  - A link type can model a many-to-many relationship. In other words, an item can be linked with multiple items.
  - Content Manager manages links separately from items, allowing for flexible application designs.
  - The semantics of a link are directional, with a source and a target, so a link can be traversed bi-directionally very efficiently.
  - A link is version-independent. It can be traversed to get the latest, a specific, or all versions of the linked document. For compound document and Web content applications, this feature supports the flexibility to specify whether linked items should retain their relationships with the existing version, or update to reflect the most recent version of the various items that make up the compound document.

  Content Manager supports the folder-contains link, which supports folder hierarchy and allows users to define additional custom link types to meet specific needs within custom applications.

  References allows a reference pointer from any component in an item hierarchy to any item of any type in the system to maintain referential integrity of item relationships by following DB2 Enterprise Server database delete rules.

- In Content Manager, applications can also define attributes as foreign keys to external DB2 Enterprise Server database tables that are not part of the Content Manager schema. This capability allows applications to associate with other DB2 Enterprise Server database applications and to help ensure referential integrity with external data.

Version Control

- Content Manager supports the storage of multiple versions of documents and parts within documents. Content Manager can create a new version when any changes occur in the document content or in its indexing attributes. Each version of a document is stored as a separate item in the system. Users can access the latest version or any version of the document by specifying the desired version number. To limit the number of versions managed in the system, administrators configure how many versions exist for a single item. Content Manager automatically deletes older versions exceeding the limit.
- The authorized administrator can determine, by item type, whether a store or update operation creates a version, modifies the latest version or prompts the user to create a version.

Search and Access

Content Manager provides the following search and access technologies that give users the ability to locate and retrieve content: parametric search, full-text search and combined parametric and full-text search.

- Parametric search lets the user locate the contents by specifying criteria based on metadata attributes.
- Full-text search allows the entry of free text or keywords as search criteria against text-indexed documents to locate documents that contain pertinent content anywhere within the body of the document.
- Combined parametric and full-text search allows users to enter both metadata attributes and full-text or keywords to expand search criteria.

Enterprise-wide Content Integration

- Content Manager provides an integrated information framework for single-point access to all heterogeneous systems of content repositories (Resource Managers).

Distributed and Hierarchical Storage Management

- Content Manager allows migration of objects from one resource manager to another. It also allows automatic object migration when business growth demands an upgrade to a new hardware platform or when a physical move warrants object migration to remote servers.
- The resource managers can be distributed in geographically dispersed locations within an enterprise for faster access to frequently referenced objects.
- In addition to traditional objects such as text documents and production images, a resource manager can also store and manage a growing spectrum of digital content—from static archives to dynamic content—including scanned images, facsimiles, Personal Computer (PC) files, Extensible Markup Language (XML), audio, video, streaming video, and web content

2.4 Scope of TOE

2.4.1 Physical Boundary

The physical boundaries of the TOE are defined by the operating environment that each component of the TOE requires for effective operation. The operating environment includes the operating system, database, cryptographic provider, web application server, and system clock used to provide the timestamp used by the TOE. The TOE is a data management system comprised of the applications required for the correct enforcement of the security functions. The TOE utilizes an underlying database (DB2 Enterprise Server) that is part of the TOE's operating environment for data storage and protection. The TOE is intended to be distributed in a closed environment which has the security mechanisms that can be used to protect the data transmission and communication between the TOE components as deemed necessary.

2.4.2 Supported Configurations

The TOE components have the software dependencies on the IT environment as described in the following table.

	Operating System	Additional Software
Client for Windows 8.4		
Windows	Windows Server 2003 Standard or Enterprise Edition *(32-bit)* or with SP1 or SP2 Windows Server 2003 R2 *(32-bit)* or with SP2 Windows XP Professional *(32-bit)* or with SP2 Windows Vista Business, Ultimate, or Enterprise Edition *(32-bit)*	Database Client if database is on a different machine: DB2 Run-Time Client V9.1 FP 3 *(32-bit)*
Enterprise Edition Server v8.4 (Library Server and Resource Manager)		
AIX	AIX® 5L 5.2 (*64-bit*) (Maintenance level 9, 10) AIX 5L 5.3 (*64-bit*) (Maintenance level 5, 6, 7) AIX 6.1	DB2 Enterprise Server Edition V9.1 FP3 *(32- or 64-bit)* Encryption Module (in the Library Server and Resource Manager): IBM Crypto for C (ICC) version 1.4.5 WebSphere Application Server V6.1.0 *(32- or 64-bit)* Fix Pack 11 or later WebSphere Application Server Network Deployment V6.1.0 *(32- or 64-bit)* Fix Pack 11 or later Database Client if database is on a different machine: DB2 Run-Time Client V9.1 FP 3 *(32-bit)*
Solaris	Solaris 9 *(64-bit)* Solaris 10 *(64-bit)*	DB2 Enterprise Server Edition V9.1 FP3 *(32- or 64-bit)* Encryption Module (in the Library Server and Resource Manager): IBM Crypto for C (ICC) version 1.4.5 WebSphere Application Server V6.1.0 *(32- or 64-bit)* Fix Pack 11 or later WebSphere Application Server Network Deployment V6.1.0 *(32- or 64-bit)* Fix Pack 11 or later Database Client if database is on a different machine: DB2 Run-Time Client V9.1 FP 3 *(32-bit)*
Linux	Red Hat Enterprise Linux AS/ES/WS 4.0 *(32- or 64-bit)* Kernel 2.6.9-5 (update 3) Red Hat Enterprise Linux AS/ES/WS 5.0 *(32- or 64-bit)* Kernel 2.6.18-8.el5xen SUSE Linux Enterprise Server 9 *(32- or 64-bit)* Kernel 2.6.5-7.97 (SP3) SUSE Linux Enterprise Server 10 SP1 *(32- or 64-bit)* Kernel 2.6.16.21-0.8-default	DB2 Enterprise Server Edition V9.1 FP3 *(32- or 64-bit)* Encryption Module (in the Library Server and Resource Manager): IBM Crypto for C (ICC) version 1.4.5 WebSphere Application Server V6.1.0 *(32- or 64-bit)* Fix Pack 11 or later WebSphere Application Server Network Deployment V6.1.0 *(32- or 64-bit)* Fix Pack 11 or later Database Client if database is on a different machine: DB2 Run-Time Client V9.1 FP 3 *(32-bit)*
z/Linux	Red Hat Enterprise Linux AS/ES/WS 4.0 *(64-bit)* Kernel 2.6.9-5 (update 3)	DB2 Enterprise Server Edition V9.1 FP3 *(32- or 64-bit)*

	Red Hat Enterprise Linux AS/ES/WS 5.0 *(64-bit)* Kernel 2.6.18-8.el5xen SUSE Linux Enterprise Server 9 *(64-bit)* Kernel 2.6.5-7.97 (SP3) SUSE Linux Enterprise Server 10 SP1 *(64-bit)* Kernel 2.6.16.21-0.8-default	Encryption Module (in the Library Server and Resource Manager): IBM Crypto for C (ICC) version 1.4.5 WebSphere Application Server V6.1.0 *(32- or 64-bit)* Fix Pack 11 or later WebSphere Application Server Network Deployment V6.1.0 *(32- or 64-bit)* Fix Pack 11 or later Database Client if database is on a different machine: DB2 Run-Time Client V9.1 FP 3 *(32-bit)*
Windows	Windows Server 2003 Standard or Enterprise Edition *(32-bit)* or with SP1 or SP2 Windows Server 2003 R2 *(32-bit)* or with SP2	DB2 Enterprise Server Edition V9.1 FP3 *(32- or 64-bit)* Encryption Module (in the Library Server and Resource Manager): IBM Crypto for C (ICC) version 1.4.5 WebSphere Application Server V6.1.0 *(32- or 64-bit)* Fix Pack 11 or later WebSphere Application Server Network Deployment V6.1.0 *(32- or 64-bit)* Fix Pack 11 or later Database Client if database is on a different machine: DB2 Run-Time Client V9.1 FP 3 *(32-bit)*
System Administration Client		
AIX	AIX® 5L 5.2 (*64-bit*) (Maintenance level 9, 10) AIX 5L 5.3 (*64-bit*) (Maintenance level 5, 6, 7) AIX 6.1	Database Client if database is on a different machine: DB2 Run-Time Client V9.1 FP 3 *(32-bit)*
Solaris	Solaris 9 *(64-bit)* Solaris 10 *(64-bit)*	Database Client if database is on a different machine: DB2 Run-Time Client V9.1 FP 3 *(32-bit)*
Linux	Red Hat Enterprise Linux AS/ES/WS 4.0 *(32- or 64-bit)* Kernel 2.6.9-5 (update 3) Red Hat Enterprise Linux AS/ES/WS 5.0 *(32- or 64-bit)* Kernel 2.6.18-8.el5xen SUSE Linux Enterprise Server 9 *(32- or 64-bit)* Kernel 2.6.5-7.97 (SP3) SUSE Linux Enterprise Server 10 SP1 *(32- or 64-bit)* Kernel 2.6.16.21-0.8-default	Database Client if database is on a different machine: DB2 Run-Time Client V9.1 FP 3 *(32-bit)*
Windows	Windows Server 2003 Standard or Enterprise Edition *(32-bit)* or with SP1 or SP2 Windows Server 2003 R2 *(32-bit)* or with SP2 Windows XP Professional *(32-bit)* SP2 Windows Vista Business, Ultimate, or Enterprise Edition *(32-bit)*	Database Client if database is on a different machine: DB2 Run-Time Client V9.1 FP 3 *(32-bit)*

2.4.3 Excluded Components

The Content Manager installation package includes the following components that are excluded from the Content Manager TOE:

- DB2 Information Integrator for Content (II4C)—a separate product delivered along with the Content Manager product package to faciliate the development of user applications..
- eClient—a browser-based web client that runs on Mozilla and Internet Explorer.

Although included in the Content Manager package, these components are not installed with Content Manager and must be installed separately. They are not required in order for Content Manager to operate and do not contribute to the security functionality provided by Content Manager. They are therefore excluded from the TOE.

Content Manager can be configured to use an LDAP directory as an external repository for user accounts. This configuration option is not covered by the Content Manager evaluation. Similarly, while Content Manager supports extendible user authentication features (via Authentication User Exits and Custom Login User Exit), these features are not included within the scope of the evaluation.

2.4.4 Logical Boundary

The logical boundaries of the TOE and the IT environment can be described in the terms of the security functions.

2.4.4.1 TOE Logical Boundary

2.4.4.1.1 Audit Function

All security-related events within Content Manager are logged. These are tied to the user/administrator that performed the action, the action performed, and the time it was performed. These audit records are stored in a central location in the IT Environment. The Client for Windows provides an interface for users to view audit records of events associated with a TOE resource.

2.4.4.1.2 Identification and Authentication

Content Manager identifies and authenticates non-administrative users before any other actions can be performed. The non-administrative user is required to provide a user name and password, which will be verified by the Library Server database table. If the verification is successful access into the TOE is granted.

2.4.4.1.3 User Data Protection

Access to the TOE's resources is governed by the resource's Access Control List (ACL) that identifies the user and the access allowed. The TOE uses privileges to define what operations a user is allowed to perform on the resources. The Library Server verifies that the user has the required privilege and the ACL associated to the requested object grants access.

2.4.4.1.4 Security Management

The System Administration Client provides the authorized administrator the capability to manage the security-related functions and attributes, such as the audit function, management of users and their associated data.

2.4.4.1.5 Protection of the TSF

Content Manager provides the mechanism used to enforce the access control policy ensuring that only authorized users are given access to the resources.

2.4.4.2 IT Environment Logical Boundary

2.4.4.2.1 Audit Function

The IT environment includes the database which stores the TSF data, including the audit records generated by the TOE. The database provides the tools used by the authorized users to review the records and to protect the data from unauthorized modifications.

2.4.4.2.2 Identification and Authentication

The underlying IT environment provides the mechanisms to identify and authenticate the authorized administrators of the TOE.

2.4.4.2.3 Cryptographic Support

The IT environment includes the IBM Crypto for C, also known as ICC ToolKit, which provides cryptographic mechanisms used by the TOE. The ICC Toolkit (specifically, version 1.4.5) is a FIPS 140-2 validated cryptographic module.

The ICC Toolkit version 1.4.5 has FIPS 140-2 validation certificate #775 and is listed on the National Institute of Standards and Technology (NIST) web site at http://csrc.nist.gov/cryptval/140-1/1401val2007.htm. The ICC toolkit is used by the Library Server to encrypt and decrypt administrator passwords that it stores. The Library Server also uses the ICC toolkit to hash user passwords and also object access tokens to ensure their integrity.

In addition, the Resource Manager component relies on cryptographic functions provided by the IBMJCEFIPS toolkit, which is part of WebSphere Application Server. The IBMJCEFIPS toolkit has FIPS 140-2 validation certificate #497 and is listed on the NIST web site at http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140val-all.htm. The IBMJCEFIPS toolkit is used by the Resource Manager to encrypt and decrypt administrator passwords that it stores.

2.4.4.2.4 Protection of the TSF

The TOE depends on the IT environment to provide the secure operating system for a real-time domain where the TOE software executes. The operating system protects the TOE and its related data against tampering and bypassing and protects data transmitted between the TOE components. The database is used to store and protect the TSF data.

3 TOE Security Environment

The TOE security environment consists of the organizational security policies and usage assumptions as they relate to TOE. The TOE provides for a level of protection that is appropriate for IT environments that require control over what information is accessed by the users on the systems. It is suitable for use in both commercial and government environments. The organizational security policies enforced by the TOE are sufficient to mitigate and counter any implied threat to the assets protected by the TOE.

3.1 Organizational Security Policies

An organizational security policy is a set of rules, practices, and procedures imposed by an organization to address its security needs. This section identifies the organizational security policies applicable to TOE and its environment.

P.OBJ_ACCESS The TOE must limit the access to, modification of, and destruction of the resource objects to those users that are authorized to access the resource object.

P.ACCOUNTABILITY Users of the system shall be held accountable for their security relevant actions within the system.

P.AUTH_USERS Only those users who have been authorized to access the information within the TOE may access the TOE.

P.MANAGE The TOE must provide authorized administrators with utilities to effectively manage the security-related functions of the system.

P.PARTIAL_SEP The TOE must maintain a domain for its own execution that protects itself and its resources from external interference, tampering or unauthorized disclosure, through its own interfaces.

3.2 Secure Usage Assumptions

This section describes the security aspects of the environment in which the TOE will be utilized. This includes information about the physical, personnel, and system aspects of the environment.

3.2.1 Physical Assumptions

A.PROTECT The TOE will be located within controlled facilities which will prevent unauthorized physical access and modification.

3.2.2 Personal Assumptions

A.AUTH_DATA Authorized users of the TOE will keep all their authentication data private.

A.MANAGE There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.

A.NOEVIL The administrative personnel are not careless, willfully negligent, or hostile and will follow and abide by the instructions provided by the administrative guidance.

3.2.3 System Assumptions

A.OS It is assumed that the operating systems have been configured in accordance with the manufacturer's installation guides and where applicable, in its evaluated configuration. It is securely configured such that the operating systems protect the TOE from any unauthorized users or processes.

A.SYSTEM The underlying operating environment will provide protection to the TOE and its stored, processed, and transmitted data, and a reliable system time.

4 Security Objectives

This section describes the security for the TOE and its supporting environment. Security objectives, categorized as either security objectives of the TOE or security objectives of the environment, reflect the stated intent to enforce the organizational security policies and address assumptions.

4.1 Security Objectives of the TOE

The following security objectives are intended to be satisfied by the TOE and its security related functions.

O.ACCOUNTABILITY
The TSF must record the security relevant actions of the users of the TOE to ensure that users are held accountable for their actions on the TOE.

O.AUTHORIZE
The TSF must ensure that only authorized users and administrators gain access to the TOE and its resources.

O.MANAGE
The TSF must allow administrators to effectively manage the TOE and its security functions, and must ensure that only authorized administrators are able to access such functionality.

O.OBJ_ACCESS
The TSF must limit access to objects maintained by the TOE to users with authorization and appropriate privileges. The TSF must allow authorized users to specify which users may access their objects and the actions performed on the objects.

O.PARTIAL_SEP
The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering or unauthorized disclosure, through its own interfaces.

4.2 Security Objective of the IT Environment

The following security objectives for the IT environment of the TOE must be satisfied in order for the TOE to fulfill its own security objectives.

OE.AUDITING
The IT environment must ensure that the audit records are available and provide the authorized user with the means to review the audit record.

OE.AUTHORIZED
The IT environment must ensure that TOE administrative users are authenticated before access to the TOE and its resources is granted.

OE.SEP
The TOE operating environment shall provide mechanisms to isolate the TSF, to ensure that TSF components cannot be tampered with or bypassed, and to protect the communication between the TOE components.

OE.TIME
The operating environment shall provide an accurate timestamp.

4.3 Security Objective of the Non - IT Environment

The following security objectives are intended to be satisfied by the environment of the TOE.

OE.CREDEN
Those responsible for the TOE must ensure that all access credentials, such as passwords are protected by the users in a manner that maintains IT security objectives.

OE.INSTALL
Those responsible for the TOE must ensure that the TOE and its operating environment is delivered, installed, managed, and operated in a manner that maintains the TOE security objectives.

OE.PERSON
Authorized users of the TOE shall be properly trained in the configuration and usage of the TOE and are trusted to follow the guidance provided for secure operation.

OE.PHYCAL Those responsible for the TOE must ensure that the parts of the TOE critical to security policy are protected from physical attack that might compromise the TOE security objectives.

5 IT Security Requirements

This section of the ST details the Security Functional Requirements (SFRs) for the TOE and the IT Environment that will support the TOE. The SFRs are a combination of the SFRs drawn from the CC Part 2 and an explicitly stated requirement that defines functionality not modeled by the CC.

5.1 TOE Security Functional Requirements

The following table lists the security functional requirements that are satisfied by the TOE.

Security Functional Class	Security Functional Requirements
FAU: Security Audit	FAU_GEN_EXP.1 Audit Data Generation
	FAU_GEN.2 User Identity Association
	FAU_SAR.1(a) Audit Review
FCS: Cryptographic Support	FCS_CKM.1(a) Cryptographic Key Generation
	FCS_CKM.4 Cryptographic Key Destruction
	FCS_COP.1(a) Cryptographic Operation
FDP: User Data Protection	FDP_ACC.2 Complete Access Control
	FDP_ACF.1 Security Attribute Based Access Control
FIA: Identification and Authentication	FIA_AFL.1 Authentication Failure Handling
	FIA_ATD.1(a), (b) User Attribute Definition
	FIA_UAU.2(a) User Authentication Before any Action
	FIA_UID.2(a) User Identification Before any Action
FMT: Security Management	FMT_MOF.1 Management of Security Functions Behaviour
	FMT_MSA.1(a), (b) Management of Security Attributes
	FMT_MSA.2 Secure Security Attributes
	FMT_MSA.3 Static Attribute Initialization
	FMT_MTD.1(a) through (f) Management of TSF Data
	FMT_SMF.1 Specification of Management Functions
	FMT_SMR.1 Security Roles
FPT: Protection of the TSF	FPT_RVM.1(a) Non-bypassability of the TSP
	FPT_SEP_EXP.1 TSF Domain Separation

Table 1: TOE Security Functional Requirements

5.1.1 Security Audit (FAU)

5.1.1.1 FAU_GEN_EXP.1 Audit Data Generation

FAU_GEN_EXP.1.1 The TSF shall be able to generate an audit record of the following auditable events:

a) All modification to the user security attributes including the authentication data

b) All modification to the objects' security attributes

c) All operations on objects

d) All attempts to log into the TOE.

FAU_GEN_EXP.1.2 The TSF shall record within each audit record at least the following information: date and time of the event; type of event; subject identity; outcome of the event; and, for operations on objects, the identity of the object.

5.1.1.2 FAU_GEN.2 User Identity Association

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 FAU_SAR.1(a) Audit Review

FAU_SAR.1(a).1 The TSF shall provide [**authorized non-administrative users**] with the capability to read [**audit records associated with actions performed on a resource**] from the audit records.

FAU_SAR.1(a).2 The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

5.1.2 Cryptographic Support (FCS)

5.1.2.1 FCS_CKM.1(a) Cryptographic key generation

FCS_CKM.1(a).1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**KDF (key derivation function)**] and specified cryptographic key sizes [**64 bits**] that meet the following: [**SAFER K-64**].

5.1.2.2 FCS_CKM.4 Cryptographic key destruction

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 value**] that meets the following: [**no standard**].

5.1.2.3 FCS_COP.1(a) Cryptographic operation

FCS_COP.1(a).1 The TSF shall perform [**hashing**] in accordance with a specified cryptographic algorithm [**Matyas-Meyer-Oseas hash function using SAFER encryption algorithm**] and cryptographic key sizes [**64 bits**] that meet the following: [**none**].

5.1.3 User Data Protection (FDP)

5.1.3.1 FDP_ACC.2 Complete Access Control

FDP_ACC.2.1 The TSF shall enforce the [**Access Control SFP**] on [**Subjects: Users; Objects: Resources**] and all operations among subjects and objects covered by the SFP.

FDP_ACC.2.2 The TSF shall ensure that all operations between any subject in the TSC and any object within the TSC are covered by an access control SFP.

5.1.3.2 FDP_ACF.1 Security Attribute Based Access Control

FDP_ACF.1.1 The TSF shall enforce the [**Access Control SFP**] on objects based on the following: [

Users:

- **User Name**
- **Groups**
- **Privileges**
- **Domain**

Resources:

- **ACL**
- **Domain**

]**.**

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [

The requested operation is allowed if:

1. **If administrative domains have been enabled, the Resource Domain must be PUBLIC or the same as the User Domain AND**
2. **The User's Privileges allow the requested operation to be performed by the User AND**
3. **(The User Privileges include the privilege to bypass ACL checking OR**
4. **Public Access is enabled and the ACL Public Rule grants the requested operation OR**
5. **The ACL includes a rule for the User Name that grants the requested operation OR**
6. **(Public Access is disabled or the ACL Public Rule does not grant the requested operation) and (the ACL does not include a rule for the User Name) and (the ACL includes a rule for a Group assigned to the User that grants the requested operation));**

Otherwise the operation fails

].

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [**no explicit authorization rules**].

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the [**no explicit deny rules**].

5.1.4 Identification and Authentication (FIA)

5.1.4.1 FIA_AFL.1 Authentication Failure Handling

FIA_AFL.1.1 The TSF shall detect when [***an administrator configurable positive integer within [1 - 32767]***] unsuccessful authentication attempts occur related to [**user logon**].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [**lock the user account, and ensures it remains locked until unlocked by an authorized administrator**].

5.1.4.2 FIA_ATD.1(a) User Attribute Definition

FIA_ATD.1(a).1 The TSF shall maintain the following list of security attributes belonging to individual **CM-defined** users: [**user name, groups, authentication data, privileges, domain, default ACL, unsuccessful authentication counter**].

5.1.4.3 FIA_ATD.1(b) User Attribute Definition

FIA_ATD.1(b).1 The TSF shall maintain the following list of security attributes belonging to individual **IT environment-defined** users: [**user name, privileges, domain**].

5.1.4.4 FIA_UAU.2(a) User Authentication Before any Action

FIA_UAU.2(a).1 The TSF shall require each **CM-defined** user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

5.1.4.5 FIA_UID.2(a) User Identification Before any Action

FIA_UID.2(a).1 The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

5.1.5 Security Management (FMT)

5.1.5.1 FMT_MOF.1 Management of Security Functions Behaviour

FMT_MOF.1.1 The TSF shall restrict the ability to [***modify the behaviour of***] the functions [**audit, access control**] to [**authorized administrator**].

5.1.5.2 FMT_MSA.1(a) Management of Security Attributes

FMT_MSA.1(a).1 The TSF shall enforce the [**Access Control SFP**] to restrict the ability to [***query, modify, delete,* [create*]***] the security attributes [**user name, groups, privileges, domain**] to [**authorized administrator**].

5.1.5.3 FMT_MSA.1(b) Management of Security Attributes

FMT_MSA.1(b).1 The TSF shall enforce the [**Access Control SFP**] to restrict the ability to [***modify***] the security attributes [**ACL**] to [**authorized administrator, authorized non-administrative user with the ‘ItemSetSysAttr’ or ‘ItemSetACL’ privilege**].

Application Note: *This refers to modifying which ACL is associated with the object, not modifying the ACL rules within the ACL. The TOE treats ACLs in the following way: ACLs are first created either by an authorized administrator (see FMT_MTD.1(c), (d)) or by an authorized non-administrative user with the ‘UserACLOwner’ privilege (see FMT_MTD.1(d)). The creator names the ACL and creates the individual ACL rules for the ACL. ACLs can be initially assigned to objects in a variety of ways (see FMT_MSA.3). Once assigned to an object, only the authorized administrator or the authorized non-administrative user assigned either the ItemSetSysAttr or ItemSetACL privilege (and granted the equivalent privilege by the ACL itself) can modify the association of an ACL to an object.*

5.1.5.4 FMT_MSA.2 Secure Security Attributes

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

5.1.5.5 FMT_MSA.3 Static Attribute Initialization

FMT_MSA.3.1 The TSF shall enforce the [**Access Control SFP**] to provide [*[**authorized administrator-specified**]*] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the [**authorized administrator, authorized non-administrative user with the 'ItemSetSysAttr', 'ItemSetACL', or 'UserACLOwner' privilege**] to specify alternative initial values to override the default values when an object or information is created.

5.1.5.6 FMT_MTD.1(a) Management of TSF Data

FMT_MTD.1(a).1 The TSF shall restrict the ability to [***modify***] the [**unsuccessful login attempts threshold**] to [**authorized administrator**].

5.1.5.7 FMT_MTD.1(b) Management of TSF Data

FMT_MTD.1(b).1 The TSF shall restrict the ability to [***modify***] the [**authentication data of other users**] to [**authorized administrator**].

5.1.5.8 FMT_MTD.1(c) Management of TSF Data

FMT_MTD.1(c).1 The TSF shall restrict the ability to [***modify, delete, [create]***] the [**administrative ACLs**] to [**authorized administrator**[i]].

5.1.5.9 FMT_MTD.1(d) Management of TSF Data

FMT_MTD.1(d).1 The TSF shall restrict the ability to [***modify, delete, [create]***] the [**user ACLs**] to [**authorized administrator, authorized non-administrative users with the 'UserACLOwner' privilege**].

5.1.5.10 FMT_MTD.1(e) Management of TSF Data

FMT_MTD.1(e).1 The TSF shall restrict the ability to [***modify***] the [**Resource domain**] to [**authorized administrator**].

5.1.5.11 FMT_MTD.1(f) Management of TSF Data

FMT_MTD.1(f).1 The TSF shall restrict the ability to [***[unlock]***] the [**user accounts**] to [**authorized administrator**].

5.1.5.12 FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions: [

a) **Management of user attributes; which includes the ability to create, modify, delete, and unlock user accounts, and modify user authentication data,**

b) **Management of object security attributes,**

c) **Ability to configure the unsuccessful login attempts threshold,**

d) **Ability to determine the events that will be logged in the event tables**

e) **Ability to enable and disable Public Access checking in access control**].

[i] Note that in this case applicable domain sub-administrators are not considered authorized administrators.

5.1.5.13 FMT_SMR.1 Security Roles

FMT_SMR.1.1 The TSF shall maintain the roles [**authorized administrator, authorized non-administrative users**].

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

Application Note: *The authorized administrator role includes a lesser role, the domain sub-administrator. The domain sub-administrator has all administrator authorities, except the ability to create, update, or delete ACLs, within their assigned domain of administration. The authorized non-administrative user is considered a security management role only in the specific cases where the user is granted 'ItemSetSysAttr', 'ItemSetACL' or 'UserACLOwner' privileges and thereby has some special security management rights.*

5.1.6 Protection of the TSF (FPT)

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

FPT_RVM.1(a).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 FPT_SEP_EXP.1 TSF Domain Separation

FPT_SEP_EXP.1.1 The TSF shall maintain a security domain that protects it from interference and tampering by untrusted subjects initiating actions through its own TSFI.

FPT_SEP_EXP.1.2 The TSF shall enforce separation between the security domains of subjects in the TSC.

5.2 IT Environment Security Functional Requirements

The following table identifies the SFRs to be satisfied by the IT environment of the TOE.

Security Functional Class	Security Functional Requirements
FAU: Security Audit	FAU_SAR.1(b) Audit Review
	FAU_SAR.3 Selectable Audit Review
	FAU_STG.1 Protected Audit Trail Storage
FCS: Cryptographic Support	FCS_CKM.1(b) Cryptographic Key Generation
	FCS_COP.1(b), (c) Cryptographic Operation
FIA: Identification and Authentication	FIA_ATD.1(c) User Attribute Definition
	FIA_UAU.2(b) User Authentication Before any Action
	FIA_UID.2(b) User Identification Before any Action
FPT: Protection of the TSF	FPT_ITT.1 Basic Internal TSF Data Transfer Protection
	FPT_RVM.1(b) Non-bypassability of the TSP
	FPT_SEP.1 TSF Domain Separation
	FPT_STM.1 Reliable Time Stamps

Table 2: IT Environment Security Functional Requirements

5.2.1 Security Audit (FAU)

5.2.1.1 FAU_SAR.1(b) Audit Review

FAU_SAR.1(b).1 The ~~TSF~~ **IT Environment** shall provide [**IT environment-defined users**] with the capability to read [**all records allowed by user privilege**] from the audit records.

FAU_SAR.1(b).2 The ~~TSF~~ **IT Environment** shall provide the audit records in a manner suitable for the user to interpret the information.

5.2.1.2 FAU_SAR.3 Selectable Audit Review

FAU_SAR.3.1 The ~~TSF~~ **IT Environment** shall provide the ability to perform [***searches, sorting***] of audit data based on [**data and time, user name, resource type**].

5.2.1.3 FAU_STG.1 Protected Audit Trail Storage

FAU_STG.1.1 The ~~TSF~~ **IT Environment** shall protect the stored audit records from unauthorized deletion.

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

5.2.2 Cryptographic Support (FCS)

5.2.2.1 FCS_CKM.1(b) Cryptographic key generation

FCS_CKM.1(b).1 The ~~TSF~~ **IT Environment** shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**Random Number Generation**] and specified cryptographic key sizes [**64 bits (for use by the KDF), 128 bits (for use by AES-CBC)**] that meet the following: [**FIPS PUB 186-2**].

5.2.2.2 FCS_COP.1(b) Cryptographic operation

FCS_COP.1(b).1 The ~~TSF~~ **IT Environment** shall perform [**encryption and decryption**] in accordance with a specified cryptographic algorithm [**AES-CBC**] and cryptographic key sizes [**128 bits**] that meet the following: [**FIPS 197 (AES)**].

5.2.2.3 FCS_COP.1(c) Cryptographic operation

FCS_COP.1(c).1 The ~~TSF~~ **IT Environment** shall perform [**hashing**] in accordance with a specified cryptographic algorithm [**SHA-1**] and cryptographic ~~key~~ **hash** sizes [**80 bits**] that meet the following: [**FIPS 180-2 (SHS)**].

5.2.3 Identification and Authentication

5.2.3.1 FIA_ATD.1(c) User Attribute Definition

FIA_ATD.1(c).1 The ~~TSF~~ **IT Environment** shall maintain the following list of security attributes belonging to individual **IT Environment-defined** users: [**user name, authentication data**].

5.2.3.2 FIA_UAU.2(b) User Authentication Before any Action

FIA_UAU.2(b).1 The ~~TSF~~ **IT Environment** shall require each **IT Environment-defined** user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

5.2.3.3 FIA_UID.2(b) User Identification Before any Action

FIA_UID.2(b).1 The ~~TSF~~ **IT Environment** shall require each **IT Environment-defined** user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

5.2.4 Protection of the TSF (FPT)

5.2.4.1 FPT_ITT.1 Basic Internal TSF Data Transfer Protection

FPT_ITT.1.1 The ~~TSF~~ **IT Environment** shall protect TSF data from [***disclosure***] when it is transmitted between separate parts of the TOE.

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

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

5.2.4.3 FPT_SEP.1 TSF Domain Separation

FPT_SEP.1.1 The ~~TSF~~ **IT Environment** shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.1.2 The ~~TSF~~ **IT Environment** shall enforce separation between the security domains of subjects in the TSC.

5.2.4.4 FPT_STM.1 Reliable Time Stamps

FPT_STM.1.1 The ~~TSF~~ **IT Environment** shall be able to provide reliable time stamps for **the TOE and** its own use.

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the components included in EAL 4 augmented with ALC_FLR.2 as specified in Part 3 of CC.

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 2: Assurance Components EAL 4 augmented with ALC_FLR.2

5.3.1 Class ACM: Configuration Management

5.3.1.1 ACM_AUT.1 Partial CM Automation

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 authorised 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 ACM_CAP.4 Generation Support and Acceptance Procedures

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 ACM_SCP.2 Problem Tracking CM Coverage

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 Class ADO: Delivery and Operation

5.3.2.1 ADO_DEL.2 - Detection of Modification

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 ADO_IGS.1 Installation, Generation, and Start-up Procedures

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 Class ADV: Development

5.3.3.1 ADV_FSP.2 Fully Defined External Interfaces

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 ADV_HLD.2 Security Enforcing High-level Design

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 ADV_IMP.1 Subset of the Implementation of the TSF

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 ADV_LLD.1 Descriptive Low-level Design

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 ADV_RCR.1 Informal Correspondence Demonstration

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.

5.3.3.6 ADV_SPM.1 Informal TOE Security Policy Model

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 Class AGD: Guidance Documents

5.3.4.1 AGD_ADM.1Administrator Guidance

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 behaviour 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 AGD_USR.1 User Guidance

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 Class ALC: Life-cycle Support

5.3.5.1 ALC_DVS.1 Identification of Security Measures

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 ALC_FLR.2 Flaw Reporting Procedures

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 ALC_LCD.1 Developer Defined Life-cycle Model

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 ALC_TAT.1 Well-defined Development Tools

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 Class ATE: Tests

5.3.6.1 ATE_COV.2 Analysis of Coverage

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 ATE_DPT.1 Testing: High-level Design

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 ATE_FUN.1 Functional Testing

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 ATE_IND.2 Independent Testing – Sample

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 Class AVA: Vulnerability Assessment

5.3.7.1 AVA_MSU.2 Validation of Analysis

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 AVA_SOF.1 Strength of TOE Security Function Evaluation

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 AVA_VLA.2 - Independent Vulnerability Analysis

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

6.1 TOE Security Functions

Each of the security function descriptions is organized by the security requirements corresponding to the security function. Hence, each function is described based on how it specifically satisfies each of its related requirements. This serves to both describe the security functions and rationalize that the security functions are suitable to satisfy the necessary requirements.

6.1.1 Audit Function

The TOE can audit system administration and item events.

System administration events comprise actions performed by an administrator in the System Administration Client or in a custom application. These events include defining users, assigning privileges, assigning access control lists to an object, and user logins. The administrator has the capability to enable or disable auditing of system administration events by modifying the Library Server configuration.

Item events are actions performed against specific objects within a Resource Manager or the object's indexing information within the Library Server. The administrator enables or disables auditing of item events by modifying an item's item type. The administrator can specify, for each item type defined in the TOE, any combination of the following actions to be audited: create; retrieve; update; and delete.

The security-related events that can be audited are as follows:

- All modification to the user security attributes including the authentication data
- All modification to the resources' security attributes
- All operations on resources
- All attempts to log into the TOE.

Each event log entry contains the following information: date and time the event occurred (obtained from the IT environment); event code (event type); item type (resource type); identity of the subject that performed the action that triggered the generation of the entry (user name); and for item events the item ID of the object acted upon. The type of event log generated and the contents of the log define the outcome of the action. (FAU_GEN_EXP.1, FAU_GEN.2)

The Library Server logs the events into event tables stored on and protected by the Library Server database located in the IT Environment. The database prevents any modification or deletions to the event tables that were not authorized by an authorized administrator. The system administration events are stored in the ICMSTSYSADMEVENTS table, and the item events are stored in the ICMSTITEMEVENTS table.

The storage allocated to the two event tables, ICMSTSYSADMEVENTS and ICMSTITEMEVENTS, limits the maximum storable volume of event data. If one or both event tables reach their capacity, subsequent attempts to log events fail and the TOE rolls back the entire action. Actions are rolled back until an administrator frees space by removing records from the event tables. To do this, the administrator uses DB2 load and unload utilities in the IT environment to preserve all or part of the tables, copy them to other media, and later restore them if needed.

The Client for Windows provides the authorized non-administrative users with the capability to view audit records for item events (events in the ICMSTITEMEVENTS table). Events related to a single item can be viewed together and are presented in a readable and understandable format. (FAU_SAR.1(a)).

In addition, users in the IT environment with appropriate access to the ICMSTSYSADMEVENTS and ICMSTITEMEVENTS tables in the Library Server database can perform SQL queries on these tables in order to search and sort the audit data by date and time, user identity, and event type. Chapter 12 of the System Administration Guide provides examples of SQL queries for extracting and reviewing audit records. The event log is presented in a readable and understandable format.

The Audit function demonstrates the implementation of the following security functional requirements: FAU_GEN_EXP.1, FAU_GEN.2, and FAU_SAR.1(a).

6.1.2 Identification and Authentication

The non-administrative users of the TOE are defined in the TOE and access the TOE via the Client for Windows or a WAS application. When a non-administrative user attempts to access the TOE, the user is identified and authenticated against the user's information stored in the user definition table of the Library Server.

The administrative users of the TOE (or superadministrators if domains have been enabled) are defined in the IT environment and access the TOE via the System Administration Client. The administrative user's user name must also be defined in the Library Server with full Content Manager administrative privileges ("AllPrivs") (FIA_ATD.1(b)). When an administrative user attempts to access the TOE, the TOE passes the information to the IT environment to identify and authenticate the user and upon successful authentication by the IT environment, the user is then identified against the user name stored in the Library Server. The user name in the Library Server must match the user name of the underlying operating system. Once the users are successfully identified and authenticated, access to the TOE and its resources is granted.

If administrative domains are defined, then users can also be "domain administrators". A domain administrator can be defined either in the TOE or in the IT environment, and the TOE manages identification and authentication of the domain administrator accordingly.

When a user logs on to the TOE, the TOE generates a User Token that represents the session under which the user can perform operations. The User Token is generated from information including the user name and the date and time (down to microseconds) the user logged on. When the user logs on, both the logon information and the User Token are returned to the API. The logon information and the User Token are then included in subsequent requests from the API to the Library Server. The Library Server can then identify the session under which the operation is being requested. The ICC Toolkit is relied on to generate SHA-1 hash values to support protection of the User Token from undetected modification (FCS_COP.1(c)).

Each user account includes a counter that tracks the number of unsuccessful attempts to authenticate into the Contact Manager. This counter is reset upon successful authentication. When the administrator-configured number of allowed attempts is exceeded, the TOE locks the user account until the authorized administrator unlocks the account by resetting the counter to zero. When the number of allowed attempts is set to zero, the user is allowed an unlimited number of unsuccessful attempts. (FIA_AFL.1)

The user definition is stored in a table on the Library Server. The information associated with CM-defined users consists of the user name, group memberships, assigned set of privileges, authentication data (password), a default ACL (that can be applied to a resource when it is created by the user), and the unsuccessful authentication counter (FIA_ATD.1(a)). The password is stored in the table hashed. The TOE uses the Matyas-Meyer-Oseas hash function with the SAFER encryption algorithm to generate the hash of the password (FCS_COP.1(a)). In addition, the Library Server component of the TOE uses the ICC Toolkit to encrypt and decrypt passwords of administrative users that it stores in the Library Server database, using 128-bit AES-CBC (FCS_COP.1(b)). Finally, the Resource Manager component of the TOE uses the IBMJCEFIPS toolkit, which is part of WebSphere Application Server in the IT environment, to encrypt and decrypt passwords of administrative users that it stores in the Resource Manager database, again using 128-bit AES-CBC (FCS_COP.1(b)).

In the evaluated configuration, the TOE is configured to invoke the ICMPLSVP user exit, supplied as part of the TOE, when a non-administrative user's password is created or modified. This user exit enforces the following password rules:

- Passwords must be at least eight (8) characters
- Passwords must contain at least 1 non-alphabetic character.

The Identification and Authentication function demonstrates the implementation of the following security functional requirements: FIA_AFL.1, FIA_ATD.1(a), FIA_ATD.1(b), FIA_UAU.2(a), FIA_UID.2(a), and FCS_COP.1(a).

6.1.3 User Data Protection

6.1.3.1 Introduction

The TOE mediates user access to controlled resources by enforcing an access control policy. The controlled resources are items and item types.

The process the TOE implements to determine if access is granted is as follows:

- If administrative domains have been enabled, the resource must either be in the PUBLIC domain or be in the same domain as the user in order for the user to be able to request an operation on the resource; and
- The user must have the appropriate privilege to perform the requested operation. If the user does not have the appropriate privilege, the user is unable to perform the requested operation on any resource; and
- The user must have appropriate authorization to access the resource with the requested operation:
    - If the user possesses the ItemSuperAccess privilege, then ACL checks are bypassed. Otherwise, the TOE then identifies the applicable binding level, which determines the ACL that will be used in the access control decision. The binding level can be Item or Item Type.
    - The ACL defines what operations the user is authorized to perform on the controlled resource. Authorization can be granted via one of three rule types, which are processed in the following order of precedence:
        - Public—authorizes the ICMPUBLC user group (i.e., all users). The performance of this check is controlled by a library server configuration parameter and is disabled by default;
        - User—authorizes identified users; and
        - Group—authorizes a user group.

      The TOE determines if the ACL authorizes the user to perform the requested operation, based on the ACL rules assigning authorizations to perform specific operations to the public, specific users, and/or specific groups.

The following sections provide further details about the various aspects of the access control policy.

6.1.3.2 Privileges

A privilege is the right to perform an action in the TOE. Privileges are classified as either administration privileges or data access privileges. Administration privileges grant rights to model user data and administer the TOE, while data access privileges grant rights to perform operations on controlled resources.

Each user is assigned a set of privileges that defines the actions the user is allowed to perform in the TOE, including operations on controlled resources (though the ACL on a controlled resource still needs to authorize the user to perform a requested action on that resource).

Privileges must be grouped into privilege sets before they can be assigned to users or specified in ACLs. Privilege sets can be defined by an authorized administrator. In addition, the TOE provides the following predefined privilege sets:

- AllPrivs – For an administrator who can perform all of the tasks described under the other privileges, including all client privileges.
- ClientUserAllPrivs – For a user who can perform all client tasks, but does not have administrator privileges. The user can search documents and perform process and folder related actions.
- ClientUserCreateAndDelete – For a user who can load documents into TOE, import and scan items, index documents, and start items on workflow and delete items.
- ClientUserEdit – For a user who can update items, annotations, and note logs, can perform searches, and can view and print documents.

- ClientUserReadOnly – For a user who can search, view, and print documents, and view annotations and note logs. The user cannot perform process related actions, folder related actions, or make any updates.
- SysAdminCM – For a TOE administrator who can perform all TOE administration tasks including managing users, privileges, and access control lists, administering the data model, and performing client tasks.
- SysAdminEIP – For a DB2 Information Integrator for Content administrator who can perform all DB2 Information Integrator for Content administration tasks including managing users, privileges, and access control lists, working with federated entities and domains, and performing all client tasks.
- SysAdminSubDomainCM – For an administrator who can work only with subdomains and users, groups, privilege sets, access control lists, and resource managers. Includes all client tasks.
- SysAdminSubDomainEIP – For a DB2 Information Integrator for Content administrator who can work only with subdomains and users, groups, privilege sets, and access control lists. Includes all client tasks.
- SysAdminSuper – For an administrator who can perform all Content Manager and Information Integrator for Content administration tasks, and all client tasks.
- UserDB2Connect – Allows users to connect to DB2 without having their own DB2 user ID. The users are required to enter a password.
- UserDB2TrustedConnect – Allows users to connect to DB2 without having their own DB2 user ID. The users do not have to enter a password. The set is comprised of the AllowConnectoLogon and AllowTrustedLogon. This privilege set is not permitted in the evaluated configuration.
- Noprivs – No privileges at all. This might be useful for a temporary user setting.

6.1.3.3 Access Control Lists

An access control list (ACL) associates one or more user IDs or user groups with privilege sets. The ACL specifies who (users, groups, or public) are authorized to perform which functions (privileges) on a controlled resource. An ACL only defines the authorization of the resource—it does not grant privileges additional to those allocated to the user.

The TOE provides the following pre-configured ACLs:

- SuperUserACL – This is an empty ACL that is assigned as the default ACL for the TOE pre-configured user (ICMADMIN).
- NoAccessACL – This ACL consists of a single rule that specifies, for all TOE users (Public), no actions (NoPrivSet) are allowed.
- PublicReadACL – This ACL consists of a single rule that specifies, for all TOE users (ICMPUBLIC), read operation (ItemReadPrivSet) is allowed.

An authorized administrator can create ACLs that can be subsequently assigned to item types and items, or a new ACL can be created at the time that an item type is created. When the item type is created, the ACL binding level is also defined. This can be either ItemType or Item. If binding level is ItemType, the ACL associated with the Item Type definition also applies to all items of that type. If binding level is Item, then each item of that Item Type will have its own ACL that will be used to determine access. The item ACL can be:

- assigned when the item is created, or
- specified to be the Item Type ACL, or
- specified to be the default ACL of the user creating the item.

When the user requests an operation on a resource, the TOE first verifies the user's privileges to determine if the user has the privilege to perform the requested operation. If the necessary privilege is not in the user's privilege set, the user is unable to perform the operation.

Otherwise, the TOE determines the level of ACL binding (Item Type or Item) and uses the ACL to determine if the user has been authorized to perform the requested action on the resource. However, if a user is assigned the ItemSuperAccess[ii] privilege, then the TOE bypasses the ACL checks.

ACLs created by administrators are termed administrative ACLs. It is also possible for users to create, modify, and delete ACLs (termed User ACLs), if the user has the 'UserACLOwner' privilege. User ACLs can be used only on items, not on item types. A user who has 'UserACLOwner' privilege and is also granted 'UserACLOwner' privilege by the ACL can modify the ACL rules and delete the ACL.

The process of checking if the ACL authorizes the requested operation is as follows:

- First, the process verifies that Public Access has been enabled. If Public Access has been enabled, the process checks the Public rule. If the privilege(s) assigned to the public allow the action, the check is successful, and the access is granted.
- If the Public Access check fails, or Public Access is disabled, the process will check against the User rules. If the rule for the user that initiated the action authorizes the user to perform the requested action, access will be granted, else access is denied.
- However, if the ACL does not define a rule for the user, then the process continues to check against the Group rule. If the user is a member of a group that has the required privileges, then access is granted, else it is denied and the process finishes.

6.1.3.4 Object Token

If the TOE (specifically, the Library Server) grants an access request, the TOE generates an Object Token and returns this to the client making the access request via the API. The Object Token is also sent to the Resource Manager responsible for the requested item. The Object Token includes the identification of the item to which access is granted, the identity of the user to whom access has been granted, the operations that have been granted, and an expiration time. The client that has successfully requested access from the Library Server submits the Object Token and the necessary access request information via a URL to the Resource Manager, which then verifies the Object Token and access request information against the Object Token received from the Library Server. In this way, the Resource Manager can determine that the access request has been granted by the Library Server and has not expired, and can return the requested item to the requesting client.

The User Data Protection function demonstrates the implementation of the security functional requirements: FDP_ACC.2 and FDP_ACF.1.

Cryptographic capabilities are provided both by the TOE and by the IT environment. The TOE implements the SAFER K-64 encryption algorithm and uses this in conjunction with the Matyas-Meyer-Oseas hash algorithm to generate a hash value used to protect the Object Token from undetected modification (FCS_COP.1(a)). In addition, the TOE implements a Key Derivation Function to derive keys for use by the SAFER K-64 algorithm, ensuring only secure values are accepted for the key values used by the Key Derivation Function (FCS_CKM.1(a), FMT_MSA.2). The TOE overwrites each (raw) key with zeros when it is no longer needed (FCS_CKM.4). TOE relies on the ICC Toolkit in the IT environment to generate random numbers to support the TOE's Key Derivation Function (FCS_CKM.1(b)). In addition, the ICC Toolkit is relied on to generate SHA-1 hash values to support protection of the User Token from undetected modification (FCS_COP.1(c)).

6.1.4 Security Management

There are four types of users associated with the TOE: authorized administrator; authorized non-administrative user; database connection user; and domain sub-administrators.

The authorized administrator user type is a TOE security management role and has administrative control of the TOE. The authorized administrator is defined as being a user assigned administration privileges (e.g., the "AllPrivs" privilege set). An authorized administrator's user ID is defined in the underlying IT environment of the TOE and must also have full database administrative privileges for the DB2 database supporting the TOE. In addition, the

[ii] ItemSuperAccess privilege is a member of several previously described privilege sets, including AllPrivs, ClientUserAllPrivs, SysAdminCM, SysAdminEIP, and SysAdminSuper.

authorized administrator’s user ID must be defined in the Library Server (and associated with the “AllPrivs” privilege set).

The authorized non-administrative user type is a TOE security management role in the case where the user is granted a privilege that allows the user to perform security management functions (e.g., ‘ItemSetSysAttr’, ‘ItemSetACL’, ‘UserACLOwner’).

The database connection user type is not a security management role, nor a user type that users are explicitly associated with. The database connection user is a shared user ID (default ICMCONCT) used by the TOE to connect authorized non-administrative users to the underlying DB2 database in the IT environment. Note that this is not an account recognized in order to access TOE services, but rather is used by the TOE to access IT environment services.

The domain sub-administrator user type falls within the scope of the authorized administrator security management role. The TOE provides an optional capability to enable “administrative domains”, which are used to limit administrative and user access to a section of the Library Server. As an example, administrative domains could be used where there is a large user base divided among many departments, or the Library Server is managed for multiple companies. An administrative domain consists of user IDs, user groups, privilege sets, ACLs, Resource Managers and collections. Resource Managers, collections, user IDs, and user groups can exist in only one domain at a time, but can be moved between domains. A domain sub-administrator is defined as being a user assigned administrative privileges within a domain (i.e., is granted the “SysAdminSubDomainCM” privilege set) and is an authorized administrator within their own domain, although they do not have the capability to create, update or delete ACLs. An authorized administrator that has administrative authority over the whole TOE (i.e., is not restricted to a domain) is also referred to in the TOE guidance documentation as a “superadministrator”.

The System Administration Client provides the interface utilized by the authorized administrator to perform the administrative functions. These functions include: (FMT_SMF.1)

- The ability to select if administrative events and which type of item events will be logged in the event tables. This capability is restricted to the authorized administrator (FMT_MOF.1)
- The ability to modify the behavior of the access control function by enabling or disabling Public Access, which determines whether or not the access control function checks for authorizations granted to the ICMPUBLC group. This capability is restricted to the authorized administrator (FMT_MOF.1)
- The ability to modify the unsuccessful login attempts threshold. This capability is restricted to the authorized administrator (FMT_MTD.1(a))
- The ability to unlock the user accounts. This capability is restricted to the authorized administrator (FMT_MTD.1(f))
- The ability to specify: the ACLs to be assigned to protected resources and as user defaults, which is restricted to the authorized administrator; and the ability to specify an alternative ACL associated to a resource when the resource is created, which is restricted to the authorized administrator and the authorized non-administrative user assigned the ‘ItemSetSysAttr’, ‘ItemSetACL’ or ‘UserACLOwner’ privilege (FMT_MSA.3)
- The ability to create, modify, and delete administrative ACLs. This capability is restricted to the authorized administrator (FMT_MTD.1(c))
- The ability to create, modify, and delete user ACLs. This capability is restricted to the authorized administrator and the authorized non-administrative user assigned the ‘UserACLOwner’ privilege (FMT_MTD.1(d))
- The ability, when administrative domains are enabled, to modify the domain that users, groups, privilege sets, ACLs, resource managers and collections belongs to. This capability is restricted to the authorized administrator (FMT_MTD.1(e))
- The ability to create, query, modify, and delete the following user security attributes: user name, group membership, privileges, domain. This capability is restricted to the authorized administrator (FMT_MSA.1(a))

- The ability to modify which ACL is associated with a resource. This capability is restricted to the authorized administrator and the authorized non-administrative user assigned the ‘ItemSetSysAttr’, ‘ItemSetACL’ or ‘UserACLOwner’ privilege (FMT_MSA.1(b))
- The ability to modify the password for any user defined in Content Manager. This capability is restricted to the authorized administrator (FMT_MTD.1(b))

The changes made through the System Administration Client are enforced by the Library Server. Additionally, for Resource Manager configuration messages (connections from the system administration client to the Resource Manager to configure Resource Manager), the TOE uses SSL and https protocol provided by the IT Environment to protect the Resource Manager configuration messages (FPT_ITT.1).

6.1.5 Protection of the TSF

Access to the administrative functions by the administrator and to resources by users is only possible if the user and administrator are successfully identified and authenticated. The TOE uses the identification and authentication mechanism and the privileges associated to the users to ensure a secure domain for the user within the TOE at its interfaces. The identification and authentication mechanism ensures that TOE interfaces are isolated to the identified user and the privileges determines what rights that user has on the TOE and what actions the user will be able to perform within the TOE. The IT environment provides the secure operating system for a real-time domain where the TOE software executes which ensures that the TOE will not be bypassed or tampered with.

The access to resources is further limited to the privileges assigned to the users and the ACL assigned to the resource. Access is only granted when the user has been authorized by the resource’s ACL. (FPT_RVM.1(a)).

The Protection of the TSF function demonstrates the implementation of the FPT_RVM.1(a) and FPT_SEP_EXP.1.

6.2 Security Assurance Measures

The following assurance measures are applied to satisfy the CC EAL4 augmented with ALC_FLR.2 assurance requirements:

- Process Assurance
- Delivery and Guidance
- Design Documentation
- Tests; and
- Vulnerability Assessment.

6.2.1 Process Assurance

6.2.1.1 Configuration Management

The Configuration Management (CM) measures applied by IBM ensure that configuration items are uniquely identified, and that the procedures documented in the configuration management documentation are used to control and track changes to the CM items. IBM ensures changes to the configuration item are properly controlled. The configuration items under CM control are the TOE implementation representation, design documentation, tests, user and administrator guidance, lifecycle documentation, vulnerability assessment, and the CM documentation:

- IBM DB2 Content Manager Enterprise Edition v8.4 Configuration Management.

The configuration management documentation satisfies:

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

6.2.1.2 Life Cycle Support

The lifecycle documentation describes the adequacy of the procedures used during the development and maintenance of the TOE through the use of a comprehensive life-cycle management plan.

The documentation describes the physical, procedural, personnel, and other development security measures that are used in the development environment to protect the TOE. It includes the physical security of the development location and any procedures used to select development staff. It further describes the procedures utilized to track all reported security flaws, the status on correcting the flaw and what measures are to be taken to correct the flaw.

- IBM DB2 Content Manager Enterprise Edition V8.4 Lifecycle document
- IBM DB2 Content Manager Enterprise Edition V8.4 Flaw Remediation.

This measure satisfies the following requirements:

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

6.2.2 Delivery and Guidance

6.2.2.1 Delivery and Installation

IBM provides documentation that explains how the TOE is delivered, the carriers utilized and the procedures that are able to maintain security when the TOE is distributed. IBM's installation procedures describe the steps used for the secure installation, generation, and start-up of the TOE along with configuration settings to secure the TOE privileges and functions.

The delivery process is documented in the IBM Delivery Operations and the installation, start-up and generation procedures are documented in:

- IBM DB2 Content Manager Enterprise Edition V8.4, Delivery, Operation and Guidance.
- IBM DB2 Content Manager Enterprise Edition Version 8 Release 4 Planning and Installing Your Content Management System.

The delivery and installation documentation satisfies the following assurance requirements:

- ADO_DEL.2
- ADO_IGS.1.

6.2.2.2 Administrative and User Guidance

IBM provides administrator guidance on how to utilize the TOE security functions, and warnings to authorized administrators about actions that can compromise the security of the TOE. The procedures included in the administrator guidance describe the steps necessary to operate the TOE in accordance with the evaluated configuration, detailing how to establish and maintain the secure configuration.

The user guidance describes the procedures to use the TOE security-related functions that are available to the non-administrative users. The procedures describe how to utilize the functions and the associated interfaces in the evaluated configuration.

The administrator guidance is documented in:

- DB2 Content Manager Enterprise Edition, DB2 Content Manager for z/OS System Administration Guide Version 8 Release 4
- DB2 Content Manager Enterprise Edition, DB2 Content Manager for z/OS Messages and Codes Version 8 Release 4

- Online helps files for the Administrator, available at:

  http://publib.boulder.ibm.com/infocenter/cmgmt/v8r4m0/index.jsp

The user guidance is documented in:

- DB2 Content Manager Enterprise Edition DB2, Content Manager for z/OS Application Programming Guide Version 8 Release 4
- Online help file for the Client for Windows (ICMClientHelpENU.zip)
- API Reference, available at:

  http://publib.boulder.ibm.com/infocenter/cmgmt/v8r4m0/index.jsp

The guidance documentation satisfies the following assurance requirements:

- AGD_ADM.1
- AGD_USR.1.

6.2.3 Design Documentation

IBM provides design documentation that includes a description of the aspects of the TOE security design, architecture and interfaces. The design documentation consists of the following:

- Functional Specification – details the interfaces and the functions of the TOE
- High-Level Design – provides a high level description of the TOE, its security functions in terms of subsystems, and describes the interfaces that communicate between the subsystems
- Low-Level Design – provides a low level description of the TOE, its security functions in terms of modules, and describes the interfaces that communicate between the modules
- Representation Correspondence – provides a mapping of the security functions and requirements to the descriptions provide in the design documentation
- Security Policy Module presents an informal security model for the TOE.

The design is described in:

- IBM DB2 Content Manager for Enterprise Edition Version 8.4 Security High-level Functional Specification and Design
- IBM DB2 Content Manager Enterprise Edition v8.4 Low-level Design Specification
- CM 84_DesignDocMapping
- IBM® DB2® Content Manager Enterprise Edition Version 8.4 Informal Security Policy Model.

The design documentation satisfies the following security assurance requirement:

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

6.2.4 Tests

The Content Manager's test documentation has been created to demonstrate appropriate breadth and depth of coverage. The test documentation describes how all security relevant functions are tested. The test documentation includes test cases and variations necessary to demonstrate that all security checks and effects related to the interfaces are correctly implemented. The test documentation provides correspondence between the security-relevant interfaces and applicable tests and test variations. The test documentation describes the procedures to successfully execute the tests, and expected results of the tests. The test documentation also includes results in the form of logs resulting from completely exercising all of the security test procedures.

The test documentation consists of the following:

- IBM® DB2® Content Manager Enterprise Edition v8.4 FP1A Security Related Test Plan
- IBM® DB2® Content Manager Enterprise Edition v8.4 FP1A Security Related Test Cases.

The test documentation satisfies the following assurance requirements:

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

6.2.5 Vulnerability Assessment

The administrator guidance documentation describes the operation of the TOE and how to maintain a secure state. The administrator guide also describes all operating assumptions and security requirements outside the scope of control of the TOE. The administrator guidance documentation has been developed to serve as a complete, clear, consistent, and reasonable administrator reference.

The strength of TOE security function analysis demonstrates that the SOF claims made in the ST for all probabilistic or permutation mechanisms are correct. IBM performs vulnerability analyses of the TOE to identify weaknesses that can be exploited in the TOE. IBM documents the status of identified vulnerabilities and demonstrates that the vulnerabilities cannot be exploited in the intended environment and that the TOE is resistant to obvious penetration attacks.

- IBM DB2 Content Manager for Enterprise Edition Version 8.4 FP1A Vulnerability Analysis
- IBM DB2 Content Manager for Enterprise Edition Version 8.4 FP1A Vulnerability Analysis Supplement Security of Function & Misuse Analysis

The vulnerability analysis documentation satisfies the following assurance requirements:

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

7 Protection Profile Claims

This TOE does not claim conformance to a Protection Profile.

8 Rationale

This section provides the rationale to demonstrate the completeness and consistency of this ST. The rationale addresses the following areas:

- Security Objectives
- Security Requirements
- Security Functional Requirement Dependencies
- TOE Summary Specification
- Strength of Function
- Internal Consistency.

8.1 Security Objectives Rationale

This section demonstrates that secure usage assumptions and organizational security policies are completely covered by security objectives. Each objective addresses or enforces at least one assumption, or organizational security policy.

Objectives	O.OBJ_ACCESS	O.ACCOUNTABILITY	O.AUTHORIZE	O.MANAGE	O.PARTIAL_SEP	OE.AUDITING	OE.AUTHORIZED	OE.SEP	OE.TIME	OE.CREDEN	OE.INSTALL	OE.PERSON	OE.PHYSAL
A.AUTH_DATA										X			
A.MANAGE										X		X	
A.NOEVIL												X	
A.OS								X			X		X
A.PROTECT											X		X
A.SYSTEM								X	X				X
P.OBJ_ACCESS	X			X									
P.ACCOUNTABILITY		X		X		X			X				
P.AUTH_USERS			X				X			X			
P.MANAGE			X	X			X						
P.PARTIAL_SEP					X								

Table 3: Policies, and Assumptions vs. Security Objectives

8.1.1 Security Objectives for the TOE

This section describes how the Security Objectives for the TOE and the Environment completely and effectively enforce the organizational policies.

O.OBJ_ACCESS This objective ensures that the TSF controls access to the objects and the actions performed on the objects managed by the TOE thus supporting the enforcement of P.OBJ_ACCESS.

O.ACCOUNTABILITY
This objective ensures that the TOE monitors, and audits the security-related actions of the users and administrators, thus supporting the enforcement of P.ACCOUNTABILITY.

O.AUTHORIZE
This objective ensures that only authorized users have access to the TOE, its functions, and the objects the TOE manages. This objective enforces P.AUTH_USERS, and supports the enforcement of P.MANAGE.

O.MANAGE
This objective ensures that the TOE provides the tools necessary for the authorized administrator to manage the security-related functions; audit function, access control function, and other administrative functions, supporting the enforcement of P.OBJ_ACCESS, P.ACCOUNTABILITY, and P.MANAGE.

O.PARTIAL_SEP
This objective ensures that the TOE maintains a domain for its own execution that protects itself and its resources from external interference, tampering or unauthorized disclosure, through its own interfaces.

8.1.2 Security Objectives for the Environment

This section demonstrates how the IT environment security objectives are effect in addressing the assumptions made about the operating environment, personnel and the physical location of the TOE.

8.1.2.1 Security Objectives for the IT Environment

OE.AUDITING
This objective ensures that the environment provides the tools required to store, and view the audit records, supporting the enforcement of P.ACCOUNTABILITY.

OE.AUTHORIZED
This objective ensures that the environment authenticates that TOE administrative users before access is granted. This supports the enforcement of P.MANAGE and P.AUTH_USERS

OE.SEP
This objective provides the support needed by the TOE to assisting in addressing A.SYSTEM and A.OS by ensuring that the TOE and it associated data cannot be tampered with or bypassed and ensuring that the data transmitted between the TOE components is protected.

OE.TIME
This objective ensures that an accurate timestamp is provided for the TOE use to accurately record information on a time/date basis, supporting A.SYSTEM and the enforcement of P.ACCOUNTABILITY.

8.1.2.2 Security Objectives for the Non-IT Environment

OE.CREDEN
This objective ensures that users of the TOE keep their authentication data (password) private. This objective supports A.AUTH_DATA, A.MANAGE and the enforcement of P.AUTH_USERS.

OE.INSTALL
This objective ensures that the TOE and its operating environment is installed, configured, managed and administered in a secure manner by a competent and security aware individual in accordance with the administrator, delivery and installation documentation for the TOE and the operating environment. This objective supports A.PROTECT and A.OS.

OE.PERSON
This objective ensures that the TOE is managed and administered in a secure manner by competent and security aware personnel in accordance with the administrator documentation. Thus, supporting A.NOEVIL and A.MANAGE

OE.PHYCAL
This objective ensures that the TOE is operated in an environment that will protect it from unauthorized access and physical threats and attacks that can disturb and corrupt the data generated and protected by the TOE. This objective addresses A.PROTECT, A.OS, and A.SYSTEM.

8.2 Security Requirements Rationale

This section demonstrates the internal consistency and completeness of the security requirements included in this ST. Table 4: Security Functional Requirements vs. Security Objectives indicates the requirements that effectively satisfy each individual objective. Objectives for the IT environment are satisfied only by requirements for the IT environment; however some of those requirements also support, indirectly, the TOE security objectives.

8.2.1 Security Functional Requirements Rationale

All SFRs identified in this ST are fully addressed in this section and each SFR is mapped to the objective for which it is intended to satisfy.

Objectives Functional Requirements	O.OBJ_ACCESS	O.ACCOUNTABILITY	O.AUTHORIZE	O.MANAGE	O.PARTIAL_SEP	OE.AUDITING	OE.AUTHORIZED	OE.SEP	OE.TIME
FAU_GEN_EXP.1		X							
FAU_GEN.2		X							
FAU_SAR.1(a)		X							
FAU_SAR.1(b)						X			
FAU_SAR.3						X			
FAU_STG.1						X			
FCS_CKM.1(a)	X								
FCS_CKM.1(b)								X	
FCS_CKM.4	X								
FCS_COP.1(a)	X								
FCS_COP.1(b)								X	
FCS_COP.1(c)								X	
FDP_ACC.2	X								
FDP_ACF.1	X								
FIA_AFL.1			X						
FIA_ATD.1(a)	X		X						
FIA_ATD.1(b)	X		X						
FIA_ATD.1(c)							X		
FIA_UAU.2(a)			X						
FIA_UAU.2(b)							X		
FIA_UID.2(a)			X						
FIA_UID.2(b)							X		
FMT_MOF.1				X					
FMT_MSA.1(a)	X		X	X					

Objectives Functional Requirements	O.OBJ_ACCESS	O.ACCOUNTABILITY	O.AUTHORIZE	O.MANAGE	O.PARTIAL_SEP	OE.AUDITING	OE.AUTHORIZED	OE.SEP	OE.TIME
FMT_MSA.1(b)	X			X					
FMT_MSA.2	X								
FMT_MSA.3	X			X					
FMT_MTD.1(a)			X	X					
FMT_MTD.1(b)			X	X					
FMT_MTD.1(c)	X								
FMT_MTD.1(d)	X								
FMT_MTD.1(e)	X								
FMT_MTD.1(f)				X					
FMT_SMF.1				X					
FMT_SMR.1			X						
FPT_ITT.1								X	
FPT_RVM.1(a)	X								
FPT_RVM.1(b)								X	
FPT_SEP.1								X	
FPT_SEP_EXP.1					X				
FPT_STM.1									X

Table 4: Security Functional Requirements vs. Security Objectives

The following text describes how each security objective is satisfied by the SFRs:

O.ACCOUNTABILITY *The TSF must record the security relevant actions of the users of the TOE to ensure that users are held accountable for their actions on the TOE.*

FAU_GEN_EXP.1 and FAU_GEN.2 define the security-related events that are auditable and the contents of the audit records, and ensure that the user that caused the event is identified in the event logged. In addition, FAU_SAR.1(a) specifies a capability for non-administrative users to review audit records associated with item audit events.

These requirements ensure the generation of audit records and that the audit records are associated to the user that caused the event.

O.AUTHORIZE *The TSF must ensure that only authorized users and administrators gain access to the TOE and its resources.*

FIA_UAU.2(a) and FIA_UID.2(a) ensure that the TOE provides an identification and authentication mechanism to authorize the users that access the TOE and its associated data.

FIA_AFL.1, FMT_MTD.1(a), and FMT_MTD.1(b), ensure that the TOE locks out a user who makes a number of unsuccessful attempts to logon and that the authorized administrator has the ability to set the number of unsuccessful login attempts. They also ensure the management of the authentication data and the ability to unlock user accounts.

FIA_ATD.1(a), FIA_ATD.1(b), FMT_MSA.1(a), and FMT_MTD.1(b) define the user identification, authentication data and privileges and restrict the management of the user attributes to the authorized administrator.

FMT_SMR.1 defines the roles that must be maintained by the TOE. These roles are the authorized administrators and authorized non-administrative users of the TOE.

These requirements work together to ensure that only authorized users have access to the TOE and the data the TOE is managing.

O.MANAGE	*The TSF must allow administrators to effectively manage the TOE and its security functions, and must ensure that only authorized administrators are able to access such functionality.*

FMT_MOF.1 ensures that the ability to manage the audit and access control functions is restricted to the authorized administrator.

FMT_MSA.1(a), FMT_MSA.1(b), FMT_MSA.3, and FMT_MTD.1(b) ensure that the management of the user’s security attributes is restricted appropriately to authorized administrators and specifically authorized non-administrative users.

FMT_MTD.1(a) ensures that the ability to modify the unsuccessful login attempts threshold is restricted to the authorized administrator.

FMT_MTD.1(f) ensures the ability to unlock user accounts is restricted to the authorized administrator.

FMT_SMF.1 ensures the authorized administrator is provided the capability to change and maintain security relevant data and functions.

These requirements work together to ensure that the security functions are restricted to the authorized administrator and the administrator has the capability to manage the TSF and the associated TSF data.

O.OBJ_ACCESS	*The TSF must limit access to objects maintained by the TOE to users with authorization and appropriate privileges. The TSF must allow authorized users to specify which users may access their objects and the actions performed on the objects.*

FDP_ACC.2 and FDP_ACF.1 define the access control SFP that the TOE uses to grant users access to the objects managed by the TOE.

FCS_CKM.1(a) and FCS_COP.1(a) define key generation and cryptographic operation requirements that support protection of the Object Token used in access control. FMT_MSA.2 ensures only secure security attributes are accepted for security attributes involved in cryptographic operations. FCS_CKM.4 ensures old key values are zeroized when no longer needed.

FIA_ATD.1(a) and FIA_ATD.1(b) define the security attributes that are associated with the non-administrative and administrative users respectively and are used by the SFP.

FMT_MSA.1(a), FMT_MSA.1(b), and FMT_MSA.3 restrict the ability to query, modify, create, and delete security attributes to authorized administrators and appropriately authorized non-administrative users and ensures that authorized administrator-specified default values are defined for the security attributes used to enforce the SFP.

FMT_MTD.1(c) restricts the ability to create, modify, and delete administrative ACLs to the authorized administrator.

FMT_MTD.1(d) restricts the ability to create, modify, and delete user ACLs to the authorized non-administrative user assigned the ‘UserACLOwner’ privilege.

FMT_MTD.1(e) restricts the ability to modify the object's domain to the authorized administrator.

FPT_RVM.1(a) ensures that the access control SFP is enforced with each request for access to an object managed by the TOE.

These requirements work together to ensure the enforcement of the SFP policies, limiting access to objects and ensuring that the ability to manage the security attributes used by the SFP is restricted to the authorized user.

O.PARTIAL_SEP	*The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering or unauthorized disclosure, through its own interfaces.*

FPT_SEP_EXP.1 ensures that the TOE will provide a secure domain for the subjects at its own interfaces.

OE.AUDITING	*The IT environment must ensure that the audit records are available and provide the authorized user with the means to review the audit record.*

FAU_SAR.1(b) and FAU_SAR.3 ensure that the IT Environment provides the records in a readable format with the capability to review, search and sort the audit records by the authorized user.

FAU_STG.1 ensures that the audit records are available for review by ensuring that the audit records are protected from unauthorized deletions.

These requirements ensure that the audit records are protected and available for review by the authorized user.

OE.AUTHORIZED	*The IT environment must ensure that TOE administrative users are authenticated before access to the TOE and its resources is granted*

FIA_UID.2(b) and FIA_UAU.2(b) ensure that the IT Environment identifies and authenticates the TOE administrative users before access to the TOE is granted. FIA_ATD.1(c) ensures the IT environment associates the necessary security attributes of user identity and authentication data with the administrative users.

This requirement ensures that the TOE and its resources are only accessible to authorized administrators for the TOE.

OE.SEP	*The TOE operating environment shall provide mechanisms to isolate the TSF, to ensure that TSF components cannot be tampered with or bypassed, and to protect the communication between the TOE components.*

FPT_ITT.1 ensures that the IT Environment protects the communication between the separate TOE components.

FPT_SEP.1 ensures that the IT Environment protects the TOE and it resources from external tampering.

FCS_COP.1(b) ensures that encryption and decryption used by the TOE to protect administrative passwords stored in the database are performed in accordance with specified cryptographic algorithms and key sizes that are compliant with FIPS 140-2.

FCS_COP.1(c) ensures that hashing used by the TOE to support protection of the User Token is performed in accordance with specified cryptographic algorithms and hash size that is compliant with FIPS 140-2.

FCS_CKM.1(b) ensures the IT Environment provides a FIPS 140-2 validated capability to generate random numbers that are subsequently used by the TOE to generate the key used in the hashing process that protects the token used to grant object access.

FPT_RVM.1(b) ensure that the IT Environment ensures that the TOE is not bypassed.

The requirements ensure that the environment protects the TOE from untrusted processes that could attempt to tamper with or bypass the TOE.

OE.TIME *The operating environment shall provide an accurate timestamp.*

FPT_STM.1 ensures that the environment provides accurate and reliable time mechanism, which may be utilized by the TOE.

This requirement ensures that the environment provides a timestamp which is accurate and reliable.

8.2.2 Security Functional Requirement Dependency Rationale

The following table identifies each security functional requirement in this ST. The table enumerates the dependencies of each requirement as specified in the CC and then identifies the requirement in this ST that satisfies each of those dependencies. Note that in some cases a dependency is satisfied by a hierarchically (as defined in the CC) greater requirement component (identified in **bold**) or by a requirement specified on the IT environment (identified in *italics*). Note that a requirement that is both a hierarchically greater component and specified on the IT environment is identified in ***bold italics***. Where a dependency is unsatisfied, rationale for not satisfying the dependency is provided following the table.

ST Requirement	CC Dependencies	ST Dependencies
FAU_GEN_EXP.1	none	*FPT_STM.1*
FAU_GEN.2	FAU_GEN.1, FIA_UID.1	FAU_GEN_EXP.1, **FIA_UID.2(a),** ***FIA_UID.2(b)***
FAU_SAR.1(a)	FAU_GEN.1	FAU_GEN_EXP.1
FCS_CKM.1(a)	(FCS_CKM.2 or FCS_COP.1), FCS_CKM.4, FMT_MSA.2	FCS_COP.1(a), FCS_CKM.4, FMT_MSA.2
FCS_CKM.4	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1), FMT_MSA.2	FCS_CKM.1(a), FMT_MSA.2
FCS_COP.1(a)	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1), FCS_CKM.4, FMT_MSA.2	FCS_CKM.1(a), FCS_CKM.4, FMT_MSA.2
FDP_ACC.2	FDP_ACF.1	FDP_ACF.1
FDP_ACF.1	FDP_ACC.1, FMT_MSA.3	**FDP_ACC.2**, FMT_MSA.3
FIA_AFL.1	FIA_UAU.1	**FIA_UAU.2(a)**
FIA_ATD.1(a)	none	none
FIA_ATD.1(b)	none	none
FIA_UAU.2(a)	FIA_UID.1	**FIA_UID.2(a)**
FIA_UID.2(a)	none	none
FMT_MOF.1	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1, FMT_SMF.1
FMT_MSA.1(a)	FMT_SMR.1, FMT_SMF.1, (FDP_ACC.1 or FDP_IFC.1)	FMT_SMR.1, FMT_SMF.1, **FDP_ACC.2**
FMT_MSA.1(b)	FMT_SMR.1, FMT_SMF.1, (FDP_ACC.1 or FDP_IFC.1)	FMT_SMR.1, FMT_SMF.1, **FDP_ACC.2**
FMT_MSA.2	ADV_SPM.1, (FDP_ACC.1 or FDP_IFC.1), FMT_MSA.1, FMT_SMR.1	See rationale.
FMT_MSA.3	FMT_MSA.1, FMT_SMR.1	FMT_MSA.1, FMT_SMR.1
FMT_MTD.1(a)	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1, FMT_SMF.1
FMT_MTD.1(b)	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1, FMT_SMF.1
FMT_MTD.1(c)	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1, FMT_SMF.1
FMT_MTD.1(d)	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1, FMT_SMF.1
FMT_MTD.1(e)	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1, FMT_SMF.1
FMT_MTD.1(f)	FMT_SMR.1, FMT_SMF.1	FMT_SMR.1, FMT_SMF.1
FMT_SMF.1	none	none
FMT_SMR.1	FIA_UID.1	**FIA_UID.2(a),** ***FIA_UID.2(b)***
FPT_RVM.1(a)	none	none
FPT_SEP_EXP.1	none	none

ST Requirement	CC Dependencies	ST Dependencies
FAU_SAR.1(b)	FAU_GEN.1	FAU_GEN_EXP.1
FAU_SAR.3	FAU_SAR.1	*FAU_SAR.1(b)*
FAU_STG.1	FAU_GEN.1	FAU_GEN_EXP.1
FCS_CKM.1(b)	(FCS_CKM.2 or FCS_COP.1), FCS_CKM.4, FMT_MSA.2	FCS_COP.1(a), *FCS_COP.1(b)*. Also see rationale.
FCS_COP.1(b)	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1), FCS_CKM.4, FMT_MSA.2	*FCS_CKM.1(b)*. Also see rationale.
FCS_COP.1(c)	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1), FCS_CKM.4, FMT_MSA.2	none (see rationale)
FIA_ATD.1(c)	none	none
FIA_UAU.2(b)	FIA_UID.1	***FIA_UID.2(b)***
FIA_UID.2(b)	none	none
FPT_ITT.1	none	none
FPT_RVM.1(b)	none	none
FPT_SEP.1	none	none
FPT_STM.1	none	none

Functional components FCS_CKM.1(b) and FCS_COP.1(b) have dependencies on FCS_CKM.4 and FMT_MSA.2. The cryptographic modules in the IT environment are FIPS 140-2 validated. Therefore, the dependencies of key destruction and secure key values are satisfied by the validation of both these modules as FIPS 140-2 compliant. In addition, FCS_COP.1(c) has dependencies on FCS_CKM.1, FCS_CKM.4 and FMT_MSA.2. However, FCS_COP.1(c) specifies requirements for an unkeyed cryptographic hash function. Therefore, the dependencies on requirements for key generation, key destruction and secure key values are not relevant to this component.

Functional component FMT_MSA.2 is specified as a TOE SFR only because it is a dependency of FCS_CKM.1(a), FCS_CKM.4 and FCS_COP.1(a). These SFRs are satisfied by cryptographic mechanisms implemented within the TOE. The relevant security attributes are the key values used in the Key Derivation Function that implements FCS_CKM.1(a). These values are secure because they are random numbers generated by the ICC Toolkit FIPS 140-2 validated cryptographic module in the IT environment. The dependencies of FMT_MSA.2 on ADV_SPM.1, (FDP_ACC.1 or FDP_IFC.1), FMT_MSA.1, and FMT_SMR.1, while nominally satisfied by this ST, are not relevant to the specific use of this component in this ST.

8.2.3 Explicitly Stated Requirements Rationale

This Security Target defines the following explicitly stated requirements: FAU_GEN_EXP.1; and FPT_SEP_EXP.1. FAU_GEN_EXP.1 is based on the CC definition of FAU_GEN.1, but has been written to specify the actual behavior of the TOE, which does not audit enabling and disabling of the audit function. FPT_SEP_EXP.1 is based on the CC definition of FPT_SEP.1, but has been written to specify the actual behavior of the TOE. The explicitly stated requirement is necessary to model the TOE’s ability to provide a security domain for its subjects at its own interfaces.

The requirements specify straight-forward functions as they exist in the TOE and are subject to evaluation using the entire set of security assurance requirements.

8.2.4 Security Assurance Requirements Rationale

This ST contains the assurance requirements from the CC EAL4 augmented assurance package. The EAL chosen is based on the statement of the security environment (assumptions, and organizational policy) and the security objectives defined in this ST. The augmentation was chosen to provide the added assurance acquired by defining flaw remediation procedures and correcting security flaws. The sufficiency of the EAL chosen (EAL4) is justified based on those aspects of the environment that have impact upon the assurance needed in the TOE. The administrative staff is conscientious, non-hostile and well trained (A.NOEVIL, A.MANAGE, and OE.PERSON) and all users of the TOE protect all access control data (i.e., password) (OE.CREDEN). The TOE is physically protected (OE.PHYCAL), and properly and securely configured (OE.INSTALL). Given these aspects, a TOE based on good commercial development and maintenance practices is sufficient. EAL4 augmented is an appropriate level of assurance for the TOE described in this ST.

8.3 TOE Summary Specification Rationale

This section in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions and security assurance measures are suitable to meet the TOE security requirements and that the collection of security functions work together to provide all of the security requirements. Table 6: Security Functional Requirements vs. Security Functions demonstrates that functions described are sufficient to substantiate the SFRs.

Table 7: Security Assurance Requirements vs. Assurance Measures provides a mapping of TOE security assurance functions to those security assurance measures that have been implemented by the developer to ensure that the TOE meets the requirements specified by CC EAL4 augmented with ALC_FLR.2.

Security Functions Functional Requirements	AUDIT	IDENTIFICATION AND AUTHENTICATION	USER DATA PROTECTION	SECURITY MANAGEMENT	PROTECTION OF THE TSF
FAU_GEN_EXP.1	X				
FAU_GEN.2	X				
FAU_SAR.1(a)	X				
FCS_CKM.1(a)			X		
FCS_CKM.4			X		
FCS_COP.1(a)		X	X		
FDP_ACC.2			X		
FDP_ACF.1			X		
FIA_AFL.1		X			
FIA_ATD.1(a)		X			
FIA_ATD.1(b)		X			
FIA_UAU.2(a)		X			
FIA_UID.2(a)		X			
FMT_MOF.1				X	
FMT_MSA.1(a)				X	
FMT_MSA.1(b)				X	
FMT_MSA.2			X		
FMT_MSA.3				X	
FMT_MTD.1(a)				X	
FMT_MTD.1(b)				X	
FMT_MTD.1(c)				X	
FMT_MTD.1(d)				X	

Security Functions Functional Requirements	AUDIT	IDENTIFICATION AND AUTHENTICATION	USER DATA PROTECTION	SECURITY MANAGEMENT	PROTECTION OF THE TSF
FMT_MTD.1(e)				X	
FMT_MTD.1(f)				X	
FMT_SMF.1				X	
FMT_SMR.1				X	
FPT_RVM.1(a)					X
FPT_SEP_EXP.1					X

Table 6: Security Functional Requirements vs. Security Functions

Assurance Measures Assurance Requirements	PROCESS ASSURANCE	DELIVERY AND GUIDANCE	DEVELOPMENT	TESTS	VULNERABILITY ASSESSMENT
ACM_AUT.1	X				
ACM_CAP.4	X				
ACM_SCP.2	X				
ADO_DEL.2		X			
ADO_IGS.1		X			
ADV_FSP.2			X		
ADV_HLD.2			X		
ADV_IMP.1			X		
ADV_LLD.1			X		
ADV_RCR.1			X		
ADV_SPM.1			X		
AGD_ADM.1		X			
AGD_USR.1		X			

Assurance Measures Assurance Requirements	PROCESS ASSURANCE	DELIVERY AND GUIDANCE	DEVELOPMENT	TESTS	VULNERABILITY ASSESSMENT
ALC_DVS.1	X				
ALC_FLR.2	X				
ALC_LCD.1	X				
ALC_TAT.1	X				
ATE_COV.2				X	
ATE_DPT.1				X	
ATE_FUN.1				X	
ATE_IND.2				X	
AVA_MSU.2					X
AVA_SOF.1					X
AVA_VLA.2					X

Table 7: Security Assurance Requirements vs. Assurance Measures

8.4 Strength of Function Rationale

The TOE minimum strength of function of SOF-medium was chosen to be consistent with the TOE. A SOF-claim is associated with the authentication mechanism described in Identification and Authentication which supports FIA_UAU.2 and the security token used for access control, FDP_ACF.1. Specifically the key size of the hash algorithm used to protect the token is subject to SOF-analysis. Additionally, since the key size is 64-bits, the evaluated configuration requires daily key change to help meet the SOF-claim.

The SOF-medium strength level is sufficient to meet the objectives of the TOE, O.AUTHORIZED, given the organizational policies the TOE and its environment must enforce, specifically P.AUTH_USERS and P.MANAGE which ensures that the TOE provides the tools used by authorized administrator to manage the TOE and that only authorized users have access to the TOE.

8.5 Internal Consistency and Support

The selected functional requirements for the TOE and IT Environment are internally consistent. All the operations performed are in accordance with the CC. The ST does not include any instances of a requirement that conflicts with or contradicts another requirement. In instances where multiple requirements apply to the same function, the requirements and their operations do not cause a conflict between each other.

The selected requirements are mutually supportive by supporting the dependencies as demonstrated in Table 5, the rationale of the suitability of the requirements to meet the objectives; the inclusion of architectural requirements, FPT_RVM.1, FPT_SEP.1, and FPT_SEP.EXP.1, to protect the TOE; the inclusion of audit requirements to detect security-related actions and the inclusion of management requirements to provide a means to properly configure and manage the other security requirements.