# Microsoft Windows Rights Management Services (RMS) Security Target

Version 1.0
9 July 2007

**Prepared for:**

**Microsoft Corporation**

One Microsoft Way
Redmond, WA 98052-6399

**Prepared By:**

**Science Applications International Corporation**

**Common Criteria Testing Laboratory**

7125 Columbia Gateway Drive, Suite 300
Columbia, MD 21046

## LIST OF TABLES

## LIST OF FIGURES

# 1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is Microsoft Windows Rights Management Services (RMS) provided by Microsoft Corporation. Windows Rights Management Services (RMS) is an information protection technology that works with RMS-enabled applications to help safeguard digital information from unauthorized use—both online and offline, inside and outside a firewall. Using Windows Server 2003 features and security technologies, including encryption, certificates and authentication, RMS helps organizations create information protection solutions. RMS provides protection of information through persistent usage policies, which remain with the information, no matter where it goes.

The Security Target contains the following additional sections:

- **Publishing** - Publishing, in the context of RMS, is the act of using an RMS enabled application to create RMS protected content and then making that RMS protected content available to another party for their consumption. A publishing license is the artifact that an author associates with each RMS protected content. RMS provides for the creation of publishing licenses in two ways: online and offline. Online publishing requires connectivity with the RMS Server in order to create a publishing license. Offline publishing refers to an author's ability to generate RMS protected content using a Client Licensor Certificate without having to connect with the RMS Server to generate the content's publishing license. The RMS enabled application, on behalf of the author, requests a Client Licensor Certificate from the RMS Server before RMS protection can be applied to a document.
- TOE Description (Section )
- Security Environment (Section 3)
- Security Objectives (Section 4)
- IT Security Requirements (Section 5)
- TOE Summary Specification (Section 6)
- Protection Profile Claims (Section 7)
- Rationale (Section 8).

## 1.1 Security Target, TOE and CC Identification

**ST Title –** Microsoft Windows Rights Management Services (RMS) Security Target

**ST Version** – Version 1.0

**ST Date** – 9 July 2007

**TOE Identification** – Microsoft Windows Rights Management Services (RMS) 1.0 SP2

**TOE Developer** – Microsoft Corporation

**Evaluation Sponsor** – Microsoft Corporation

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

## 1.2 Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Requirements, Version 2.3, August 2005.

  - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Requirements, Version 2.3, August 2005.
  - Part 3 Conformant
  - Assurance Level: EAL 4 augmented with ALC_FLR.3
  - Strength of Function Claim: SOF-Medium

## 1.3 Conventions and Terminology

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 placed at the end of the component. For example FDP_ACC.1a and FDP_ACC.1b indicate that the ST includes two iterations of the FDP_ACC.1 requirement, a and b.
  - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
  - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
  - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

### 1.3.1 Terminology

- **RMS-enabled application** - Required to create and publish RMS-protected content. Applications can be developed specifically for RMS, or existing applications can be rewritten to work with RMS.
- **RMS machine certificate** - Identify a particular computer as trusted by RMS.
- **Lockbox (also known as security processors)** - Contain the computer's private key and a matching certificate, which contains the computer's public key.
- **Rights Management Account Certificate (RAC)** - Identify a particular user as trusted by RMS.
- **Client Licensor Certificates (CLC)** - Allows a user to publish RMS-protected content while they are disconnected from the network.
- **Server Licensor Certificate (SLC)** – Identifies a particular Root Certification Server or Licensing Server as Trusted by RMS.
- **Publishing license** - Define usage rights for a piece of content; where each license contains the content encryption key encrypted with the RMS server's public key.
- **Use license** - Allow a user to consume RMS-protected content; where each license contains the content encryption key encrypted with the authorized user's public key.
- **Publishing** - Publishing, in the context of RMS, is the act of using an RMS enabled application to create RMS protected content and then making that RMS protected content available to another party for their

consumption. A publishing license is the artifact that an author associates with each RMS protected content. RMS provides for the creation of publishing licenses in two ways: online and offline. Online publishing requires connectivity with the RMS Server in order to create a publishing license. Offline publishing refers to an author's ability to generate RMS protected content using a Client Licensor Certificate without having to connect with the RMS Server to generate the content's publishing license. The RMS enabled application, on behalf of the author, requests a Client Licensor Certificate from the RMS Server before RMS protection can be applied to a document.

# 2. TOE Description

The Target of Evaluation (TOE) is the RMS Server component of Microsoft Windows Rights Management Services (RMS) 1.0 with SP2. RMS is a set of web and operating system services designed to facilitate the management of rights-protected content. While the TOE doesn't actually store any protected content, it generates certificates and licenses that can be used to encrypt content and enable access to those authorized to use the content. RMS provides the setup steps that enable trusted entities to use rights-protected information. It also handles administration functions.

## 2.1 TOE Overview

The TOE issues XML-based licenses that define usage rights and conditions to control access to encrypted data. The TOE is supported on Windows Server 2003 as its IT environment. Encrypted data usage rights and conditions that are defined within licenses identify individual authorized users who can view the information and how that information can be used and shared.

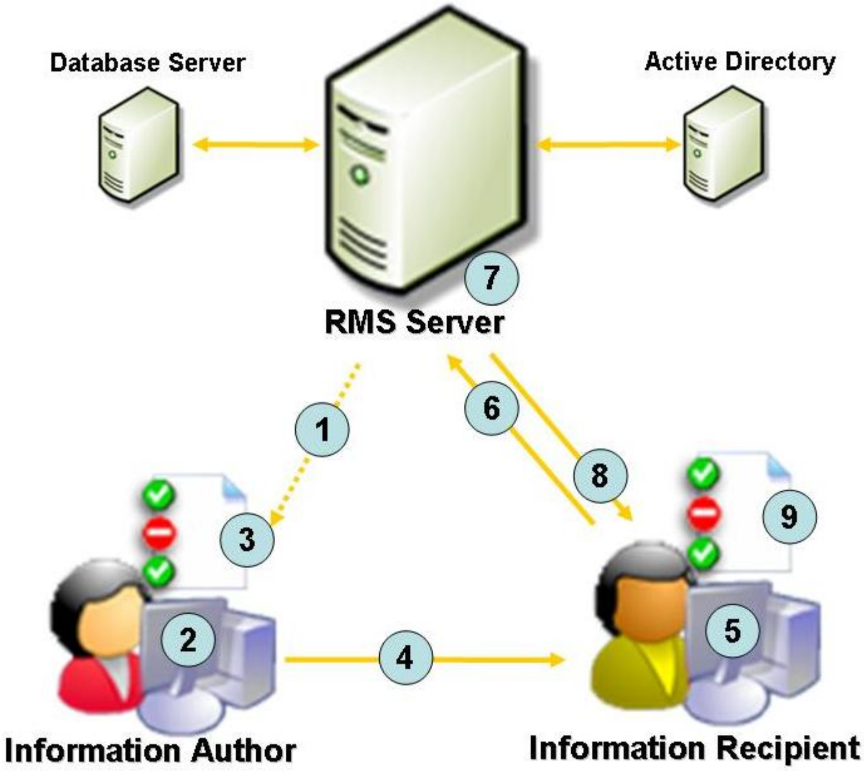

**Figure 1 Workflow of creating and viewing rights-protected information**

**Figure 1** depicts a standard workflow for RMS, but keep in mind that the TOE only includes the RMS Server and as such only directly enforces the parts of the workflow directly within its control. This process includes the following steps:

1. Author identifies to the RMS server and receives a Rights Account Certificate (RAC). Using RAC, author receives a Client Licensor Certificate (CLC) from the RMS server the first time they initiate an action on rights-protect information. This is a one-time step that enables offline publishing of rights-protected information in the future.

2. Using an RMS-enabled application, an author creates a file and defines a set of usage rights and conditions for that file. A publishing license is then generated that contains the usage policies.

3. The application encrypts the file with a symmetric key which is then encrypted to the public key of the author's Windows RMS server. A publishing license is then generated that contains the usage policies and the key. The publishing license is then bound to the file. Only the author's Windows RMS server can issue use licenses to decrypt this file.

4. The author distributes the file.

5. A recipient receives a protected file through a regular distribution channel and opens it using an RMS-enabled application or browser.

6. If the recipient does not have an account certificate on the current computer, this is the point at which one will be issued.

   The application sends a request for a use license to the RMS server that issued the publishing license for the protected information. The request includes the recipient's account certificate (which contains the recipient's public key) and the publishing license (which contains the symmetric key that encrypted the file). Note: A publishing license issued by a client licensor certificate includes the URL of the server that issued the certificate. In this case, the request for a use license goes to the Windows RMS server that issued the client licensor certificate and not to the actual computer that issued the publishing license.

7. The Windows RMS licensing server validates that the recipient is authorized, checks that the recipient is a named user, and creates a use license.

   During this process, the server decrypts the symmetric key using the private key of the server, re-encrypts the symmetric key using the public key of the recipient, and adds the encrypted session key to the use license. This step ensures that only the intended recipient can decrypt the symmetric key and thus decrypt the protected file. The server also adds any relevant conditions to the use license, such as the expiration or an application or operating system exclusion.

8. When the validation is complete, the licensing server returns the use license to the recipient's client computer.

9. After receiving the use license, the application examines both the license and the recipient's account certificate to determine whether any certificate in either chain of trust requires a revocation list. If so, the application checks for a local copy of the revocation list that has not expired. If necessary, it retrieves a current copy of the revocation list. The application then applies any revocation conditions that are relevant in the current context. If no revocation condition blocks access to the file, the application renders the data, and the user may exercise the rights they have been granted.

## 2.2 TOE Architecture

IT environment components will reside on either the local host for the root RMS certification server or RMS licensing server or on the local network infrastructure serviced by RMS. IT environment components residing on the RMS Server host operating system include IIS 6.0, Microsoft SQL Server 2005, Microsoft Message Queuing (MSMQ), ASP.NET 1.1, and Internet Explorer 6.0. IT environment components supporting RMS from the local network infrastructure include Windows Server 2003 Domain Controllers with Active Directory, Windows Server 2003 Certificate Servers, and other network services such as DNS. Note that the Windows Firewall on the local RMS hosts can be used to further secure the host and to prevent external access to SQL ports.

The following architecture component configurations will be included in the RMS evaluation:

- **Microsoft Hosted Services for RMS.** The RMS TOE must not have direct connectivity to external network environments, such as the Internet. However, RMS requires the use of RMS Enrollment service hosted by Microsoft in order to receive the root RMS server licensor certificate. In order to maintain an isolated environment for the evaluation of RMS, an offline enrollment process will be used. Note that the enrollment service is used to sign the public key of the requesting RMS server; it does not issue public/private key pairs to the RMS server. All root RMS server licensor certificate requests will be made from a computer that has Internet connectivity, but is maintained outside of the TOE. The certificate can then be imported from removable media to the RMS TOE. The ability to make root RMS server licensor certificate requests from an external computer and then import them into the root RMS server located on a separate network is a new feature that is available in the TOE.

- **RMS Root Certification Server.** The RMS evaluation will include one root RMS certification server per Active Directory forest. To support RMS, the host operating system for the root RMS certification server will also host Internet Information Services (IIS) 6.0, MSMQ, ASP.NET 1.1, Internet Explorer 6.0, and Microsoft SQL Server 2005. IIS will be configured to use SSL for the RMS Web site. The RMS evaluation will not include root RMS certification server clustering configurations.

- **RMS Licensing Server.** Within the TOE, administrators will have the option of performing server sub-enrollment to allow additional servers within the TOE evaluated configurations to issue licenses that will be trusted by the RMS system under evaluation. These RMS Licensing Servers will have IIS 6.0, configured to use SSL, MSMQ, ASP.NET 1.1, Internet Explorer 6.0, and Microsoft SQL Server 2005 locally installed. The RMS evaluation will not include RMS Licensing server clustering configurations.

- **RMS Clients.** RMS client host operating systems will include Windows Server 2003 SP2 and Windows XP Professional SP2, but are not included in the TOE. Within the RMS TOE, administrators will have the option of requiring or allowing clients to use smart cards to authenticate with the RMS server and obtain rights account certificates and licenses. Windows Rights Management Client software and RMS-enabled applications, such as Microsoft Office 2007, are not included in the TOE, but will be included in testing to verify RMS server functions.

- **Network Infrastructure Components.** Network infrastructure components will support the operating environment for the RMS evaluation, but will not be included in the TOE. RMS servers will need to be joined to a Windows domain in order to use Active Directory Services. Other services that may be used to support the network infrastructure for RMS include DHCP for clients, IP address assignments and DNS. A Windows Server 2003 Certificate Server configured in its Common Criteria evaluated configuration (see http://niap.bahialab.com/cc-scheme/st/ST_VID4024.cfm) may be used to provide smart card certificates for RMS clients and SSL certificates for IIS 6.0 RMS Web sites.

### 2.2.1 Physical Boundaries

The RMS TOE consists of the following ASP.NET based web services or applications running within the context of a Windows Server 2003 IIS application pool:

- **RMS Administration Service** – Allows the administrator to manage RMS.

- **RMS Subenrollment Service (Enrollment Service)** – Provides subordinate server licensor certificates to licensing servers. These certificates allow the licensing servers to issue publishing and use licenses.

- **RMS Certification Service** – Provides rights account certificates to users. These certificates are required for users to obtain publishing and use licenses to author and consume RMS-protected content.

- **RMS Activation Proxy Service (Activation Service)** – This service is retained for compatibility with the RMS version 1 client. It passes machine activation requests to the Microsoft Activation Service and returns lockboxes and RMS machine certificates to RMS version 1 clients. This service is not used by clients running RMS client for Service Pack 1 (SP1) or later.
- **RMS Licensing Service –** Issues use licenses, which allow users to consume RMS-protected content.
- **RMS Publishing Service** – Issues publishing licenses, which allow authors to create and distribute RMS-protected content. Also issues client licensor certificates, which allow users to publish RMS-protected content without being connected to the RMS-enabled network.
- **RMS Server Service** – The Server service exposes a request that is made by a client to retrieve a server licensor certificate chain.
- **RMS Service Locator Service** – The Service Locator service provides the service connection point URL of the server to Active Directory so that it can be discovery by RMS-enabled clients.
- **RMS Remote Active Directory Services** – This Web application supports Active Directory group membership queries.
- **RMS Decommissioning Service** – Makes previously RMS-protected content unprotected and returns it to the client. This service is installed by RMS Setup, but the service does not have a corresponding virtual root in IIS until it is enabled by the administrator. Enabling this service disables all other services.
- **RMS Logging Listener Service (Logging Service) –** Each Web service sends logged data to the logging message queue. The logging listener service then transfers the logged data from the message queue to the logging database.
- **RMS Group Expansion Service –** Determines whether a principal is a member of a group in a multiple forest Active Directory environment whose Universal group membership are not replicated among forests.

The TOE relies on the following Common Criteria related security functions from its IT environment:

- **Security Audit.** The IT Environment stores and protects the audit logs generated by the TOE in the local SQL server database. This IT environment protection is provided by the NTFS file system file permissions of the environment. It also can make them accessible to authorized users via local SQL queries only.
- **Cryptographic support**. The IT Environment generates asymmetric key pairs for use in signing and verifying signatures of rights account, server licensor, and client licensor certificates. Also, the IT Environment generates symmetric keys for use in encrypting and decrypting content.
- **Identification and authentication**. The IT Environment authenticates users using mechanisms supported by the operating system in the IT environment.
- **Security management**. The IT Environment provides the authorized administrator role and restricts access to security management functions to that administrator.
- **TSF protection**. The IT Environment ensures that the TOE is not bypassed and that it is not subject to tampering. The IT Environment also provides a reliable source of time information for audit logs.

From a non-security perspective, the TOE requires an execution environment and resources since it operates as a set of services in the context of the hosting operating system. It also requires the services of Internet Information Services (IIS) 6.0, MSMQ, ASP.NET 1.1, and Microsoft SQL Server 2005 to present its web interfaces and to store TSF data, respectively.

### 2.2.2 Logical Boundaries

This section summarizes the security functions provided by RMS:

- Security Audit,
- User data protection,
- Identification and authentication, and
- Security management.

#### 2.2.2.1 Security Audit

RMS has the ability to log use license requests. When logging is enabled, all attempts to acquire use licenses are logged by forwarding them to the local SQL server configured in the IT environment of the TOE.

#### 2.2.2.2 User data protection

The TOE ensures that certificates are generated with appropriate contents. The TOE also restricts the issuance of use licenses to content users who have been granted rights that would be reflected in a license issued by the TOE.

#### 2.2.2.3 Identification and authentication

While the TOE depends upon the IT environment to properly authenticate user identities, the TOE requires the identity of the applicable users before it can perform any of its functions.

#### 2.2.2.4 Security management

The TOE provides the administrator with functions to manage the audit function, use license issuance controls and exclusion list, the decommissioning service, and dictating the applicable content of certificates, and licenses.

### 2.2.3 Product Exclusions

The following features/capabilities of the RMS product are outside the TOE Scope of Control (TSC) and are therefore excluded from the TOE for purposes of this evaluation.

- Lockboxes (RMS Activation Service)
  Lockboxes refer to the RMS client software. RMS client software is outside the TSC because it is implemented as a user mode client side library that is potentially by-passable. Prior to the release of the client software for RMS 1.0 with SP1, lockboxes were not self-activated at the time of installation and necessitated downloading the lockbox from the Microsoft-hosted Activation Service via RMS Activation Service running on the Root Certification Server. Since all client computers are assumed to be installed with RMS 1.0 with SP2 client software, this functionality is outside the TOE.
- Pre-Licensing
  The concept of “pre-licensing” refers to the publisher requesting a Use License on behalf of another user at publishing time so that it can provide this end user with all the required licenses to consume a piece of content. This optional feature depends upon the RMS enabled client application. An RMS enabled client application that requests a Use License at publishing time has made a conscious decision to use an unevaluated capability of the TOE. Since it is assumed that all users are authenticated by the IT Environment, this functionality would violate this assumption; hence it is outside the TOE.
- Server certification
  Server certification provides a Rights Account Certificate for a server or service on a particular computer rather than an authenticated user. This optional feature is also known as “server lockbox”; it is not included in the TOE because it is assumed only authenticated users using desktop machines are consumers of RMS protected content. This capability is not accessible by default.
- Mobile device certification
  Mobile device certification provides a Rights Account Certificate for RMS clients running Windows Mobile. This optional feature is not included in the TOE because the Windows Mobile platform is not conformant with the requirements for the IT environment. This capability is not accessible by default.
- Temporary RACs
  Temporary RACs are intended for operating environments which offer anonymous or guest accounts; hence this capability is not included in the TOE. An RMS enabled client application that requests temporary RACs has made a conscious decision to use an unevaluated capability of the TOE.

- RACs based on .NET Passport credentials
  RACs based on .NET Passport credentials are outside the TOE because the mechanism used to authenticate .NET Passport credentials is not included in the evaluated configuration of the underlying IT environment.

- Group expansion across forests
  This functionality is not required because the IT Environment assumes a closed environment for a particular domain/forest when installing the RMS Server.

- Trusted domains
  The concept of "trusted domains" refers to other RMS installations that are trusted "user" and/or "publishing" sites. This optional feature is not included in the TOE because the IT Environment assumes a closed environment.

- Super users group
  The concept of a "super users" group is outside the TOE because it by-passes the RMS Use License Access Control Policy. This feature is disabled by default.

- Offline publishing (where an RMS-enabled client application issues a Publishing License)
  A publishing license created using the offline publishing model is outside the TOE because license creation is performed by the RMS client software which is outside the TSC.

- Re-Publishing
  The concept of "re-publishing" refers to modifying a Publishing License. This optional feature depends upon the RMS enabled application. It is outside the TOE because someone other than the RMS protected content's author is able to modify the Publishing License. This capability is not accessible by default.

- Revocation lists
  Revocation lists are not included in the TOE because revocation is only enforced by the RMS client software which is outside the TSC.

## 2.3 TOE Documentation

The following administrative guidance documents will be developed for the RMS evaluation:

- **Microsoft Windows Rights Management Services (RMS) Security Configuration Guide.** This document will provide installation and configuration procedures for the secure implementation of Microsoft Windows RMS server in a manner that enforces compliance with RMS SFRs.

- **Microsoft Windows Rights Management Services (RMS) Evaluated Configuration Administrator's Guide.** This document will provide all administrative guidance needed to securely operate the RMS TOE in accordance with the administrative requirements stated in the corresponding ST.

- **Microsoft Windows Rights Management Services (RMS) Evaluated Configuration User's Guide.** This document describes the functions and interfaces available to the non-administrative users, as well as the use of user-accessible security functions that are supported by the RMS TOE for RMS clients.

Refer to Section 6 for information about these and other documentation associated with Windows Rights Management Services.

# 3. Security Environment

The TOE is designed to offer Rights management Services with assurance commensurate with EAL 4 augmented with ALC_FLR.3. The following subsections define the specific threats to be addressed by the TOE and its environment as well as assumptions about the intended environments.

## 3.1 Threats

T.ACCESS The TOE may provide information inappropriately or fail to provide the correct information in response to a valid request.

T.CRYPTO The TOE may fail to use appropriate cryptographic services to support its cryptographic operations.

T.IMPERSONATE The TOE may fail to properly identify a user making a request for service.

T.MANAGE The TOE may fail to provide the functions necessary to effectively manage its own security functions.

T.PROTECT The TOE may be bypassable or subject to tampering by untrusted users.

T.UNDETECTED_ACTIONS
An unauthorized user may perform unauthorized actions that go undetected because of the failure of the system to record actions.

## 3.2 Assumptions

A.COOP Authorized users possess the necessary authorization to access at least some of the information managed by the TOE and are expected to act in a cooperating manner in a benign environment.

A.LOCATE The processing resources of the TOE will be located within controlled access facilities which will prevent unauthorized physical access.

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

A.NO_EVIL_ADM The system administrative personnel are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the administrator documentation.

A.PEER Any other systems with which the TOE communicates are assumed to be under the same management control and operate under the same security policy constraints.

A.PROTECT The TOE hardware and software critical to security policy enforcement will be protected from unauthorized physical modification.

# 4. Security Objectives

The following security objectives for the TOE and its environment are intended to counter the threats and reflect the assumptions identified for the intended TOE environment.

## 4.1 Security Objectives for the TOE

O.CONTENT The TOE must ensure that certificates and licenses have appropriate content when issued to users and are only issued when appropriate.

O.IDENTIFICATION

The TOE must be designed to ensure that it can identify each user before providing client licensor certificates or use licenses.

O.MANAGE The TOE must provide administrator functions suitable to manage the security functions of the TOE.

O.REPORTING

The TSF must record use license requests.

## 4.2 Security Objectives for the IT Environment

O.ADMIN The IT environment is responsible to ensure that an administrator role is established for the management of the TOE.

O.AUDITSTORE

The IT Environment is responsible to store and make accessible audit records generated by the TOE.

O.AUTHENTICATION

The IT Environment is responsible to authenticate users before allowing their requests to be passed to the TOE.

O.CRYPTO The IT environment is responsible to provide appropriate cryptographic functions in support of the cryptographic operations of the TOE.

O.PROTECT The IT environment is responsible to ensure that the TOE is not bypassed and is not subject to tampering.

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

## 4.3 Security Objectives for the Environment

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

O.INSTALL Those responsible for the TOE must ensure that the TOE is delivered, installed, managed, and operated in a manner which maintains IT security objectives.

O.PHYSICAL Those responsible for the TOE must ensure that those parts of the TOE critical to security policy are protected from physical attack which might compromise IT security objectives.

# 5. IT Security Requirements

This section defines the security functional and assurance requirements satisfied by Windows RMS Services. Each of these requirements has been drawn from version 2.3 of the Common Criteria, Parts 2 and 3 and has been completed in this Security Target as necessary.

## 5.1 TOE Security Functional Requirements

The following table describes the SFRs that are candidates to be satisfied by RMS.

| Requirement Class | Requirement Component |
|---|---|
| **FAU: Security Audit** | FAU_GEN_EX.1: Reporting Data Generation |
| **FDP: User data protection** | FDP_ACC.1: Subset access control |
| | FDP_ACF.1: Security attribute based access control |
| | FDP_RMS_EX.1: Certificate Generation |
| | FDP_RMS_EX.2: License Generation |
| **FIA: Identification and authentication** | FIA_UID.1: Timing of identification |
| **FMT: Security management** | FMT_MOF_EX.1: Management of validity periods |
| | FMT_MSA.3: Static attribute initialization |
| | FMT_SMF.1: Specification of Management Functions |

**Table 1 TOE Security Functional Components**

### 5.1.1 Security Audit (FAU)

#### 5.1.1.1 Reporting data generation (FAU_GEN_EX.1)

**FAU_GEN_EX.1.1** The TSF shall be able to generate a log entry of the following event: use license requests.

**FAU_GEN_EX.1.2** The TSF shall record within each log entry at least the following information: HostMachineName, HostMachineRequestId, RequestTime, RequestPath, RequestType, RequestUserAddress, RequestUserAgent, AuthenticatedState, SecureConnectionState, AuthenticatedId, ReceivedXrML, IssuedXrML, SuccessOrFailure, Metadata, and ErrorInformation.[1]

### 5.1.2 User data protection (FDP)

#### 5.1.2.1 Subset access control (FDP_ACC.1)

**FDP_ACC.2.1** The TSF shall enforce the [**RMS Use License Access Control Policy**] on [

**(a) subjects: content users,**
**(b) objects: RMS-protected content objects, and**
**(c) operations: use license requests**].

#### 5.1.2.2 Security attribute based access control (FDP_ACF.1)

**FDP_ACF.1.1** The TSF shall enforce the [**RMS Use License Access Control Policy**] to objects based on the following: [

**(a) the user identity and group membership(s) associated with a subject**

[1] See section 6.1.1 for a summary of the audit log fields.

**(b) the following publishing license access control attributes associated with a RMS-protected content object:**

**(1) The identity of the content that the license is granting rights to;**
**(2) The rights that are granted to the content;**
**(3) The identity of one or more users or groups that the license is granting rights to;**
**(4) The identity of the user that issued the license;**
**(5) A validity period is identified; and**
**(6) The symmetric content key for decrypting the content – the symmetric content key is encrypted with the server's public key**].

**FDP_ACF.1.2** The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [**A use license request operation is allowed if the following conditions are true:**

**(a) the requesting subject's user Account certificate is valid,**
**(b) the publishing license associated with the RMS-protected content object is valid,**
**(c) the requesting subject or one of its groups is identified in the publishing license, and**
**(d) the TSF successfully decrypts the symmetric content key by server private key**].

**FDP_ACF.1.3** The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [**the decommissioning service is enabled and accessible**].

**FDP_ACF.1.4** The TSF shall explicitly deny access of subjects to objects based on the [**requesting subject has been explicitly excluded for obtaining any use license**].

### 5.1.2.3 Certificate Generation (FDP_RMS_EX.1)

**FDP_RMS_EX.1.1** The TSF, when generating certificates (excludes SLC), shall verify that the prospective certificate subject is not identified by an explicit exclusion.

**FDP_RMS_EX.1.2** The TSF validates, at each level of the certificate chain, the certificate and then verifies that it connects to a known root of trust through a chain of trust. Each certificate that is in the chain is checked by the TSF to validate the following conditions:

(a) The issuer signature is valid;
(b) The actual use of the certificate matches the intended use;
(c) The validity dates are met; and
(d) The signature key and certified issuer key match.

**FDP_RMS_EX.1.3** If the TSF generates rights account certificates, at a minimum, the TSF shall ensure that:

(a) A user is identified;
(b) The user's public key is included;
(c) A specific computer is identified;
(d) A validity period is identified; and
(e) The identity of the instance of the TOE that issued the certificate is identified.

**FDP_RMS_EX.1.4** If the TSF generates server licensor certificates, at a minimum, the TSF shall ensure that:

(a) A server is identified;
(b) The server's corresponding public key is included;
(c) A validity period is identified; and
(d) The identity of the instance of the TOE that issued the certificate is identified.

**FDP_RMS_EX.1.5** If the TSF generates client licensor certificates, at a minimum, the TSF shall ensure that:

(a) A user is identified;
(b) The user's corresponding public key is included;
(c) The user's corresponding private key is included;
(d) The user's corresponding private key is encrypted using the rights account certificate public key of the client licensor requestor;
(e) A validity period is identified; and
(f) The identity of the instance of the TOE that issued the certificate is identified.

#### 5.1.2.4 License Generation (FDP_RMS_EX.2)

**FDP_RMS_EX.2.1** The TSF, when generating licenses, shall validate, at each level of the certificate chain, the license and then verify that it connects to a known root of trust through a chain of trust. Each license that is in the chain is checked by the TSF to validate the following conditions:

(a) The issuer signature is valid;
(b) The actual use of the license matches the intended use;
(c) The validity dates are met; and
(d) The signature key and certified issuer key match.

**FDP_RMS_EX.2.2** If the TSF generates publishing licenses in 'online' publishing mode, at a minimum, the TSF shall ensure that:

(a) The identity of the content that the license is granting rights to;
(b) The rights that are granted to the content;
(c) The identity of one or more users that the license is granting rights to are identified;
(d) The identity of the instance of the TOE that issued the license is identified;
(e) A validity period is identified; and
(f) The symmetric content key for decrypting the content that is encrypted with the server's public key.

**FDP_RMS_EX.2.3** If the TSF generates use licenses, at a minimum, the TSF shall ensure that:

(a) A user is identified;
(b) The symmetric content key for decrypting the content that is encrypted with the user's rights account certificate public key of the use license requestor;
(c) The identity of the instance of the TOE that issued the license is identified;
(d) A validity period is identified; and
(e) The subject requesting the use license has been authorized by the RMS License Access Control Policy.

### 5.1.3 Identification and authentication (FIA)

#### 5.1.3.1 User identification before any action (FIA_UID.1)

**FIA_UID.1.1** The TSF shall allow [**all TSF-mediated actions except for use license requests and client licensor certificate requests**] on behalf of the user to be performed before the user is identified.

**FIA_UID.1.2** The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

### 5.1.4 Security management (FMT)

#### 5.1.4.1 Management of validity periods (FMT_MOF_EX.1)

**FMT_MOF_EX.1.1** The TSF shall require the Administrator to specify the set of acceptable values for the validity period for issued certificates and licenses and usage rights templates.

#### 5.1.4.2 Static attribute initialization (FMT_MSA.3)

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

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

#### 5.1.4.3 Specification of Management Functions (FMT_SMF.1)

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

**(a) manage certificate and license validity periods,**
**(b) enable and disable the security audit function,**
**(c) enable the decommissioning service,**

**(c) manage RMS Use License Access Control Policy security attributes, including explicit exclusions**].

## 5.2 IT Environment Security Functional Requirements

The following table describes the SFRs that are to be satisfied by the IT environment of RMS. Note that the intention is that the TOE will operate in the context of Windows Server 2003 SP2 that has been evaluated against a superset of the following requirements. While those other requirements are important, this section identifies only those that have a direct bearing on the security functions of the RMS TOE.

| Requirement Class | Requirement Component |
|---|---|
| **FAU: Security audit** | FAU_STG.1: Protected audit trail storage |
| | FAU_SAR.1: Audit review |
| **FCS: Cryptographic support** | FCS_CKM.1a: Cryptographic key generation |
| | FCS_CKM.1b: Cryptographic key generation |
| | FCS_COP.1: Cryptographic operation |
| **FIA: Identification and authentication** | FIA_UAU.2: User authentication before any action |
| **FMT: Security management** | FMT_MOF.1: Management of security functions behaviour |
| | FMT_MSA.1: Management of security attributes |
| | FMT_MSA_EX.1: Management of RMS services |
| | FMT_SMR.1: Security roles |
| **FPT: Protection of the TSF** | FPT_RMS_EX.1: Protection of RMS services |
| | FPT_RVM.1: Non-bypassability of the TSP |
| | FPT_SEP.1: TSF domain separation |
| | FPT_STM.1: Reliable time stamps |

**Table 2 IT Environment Security Functional Components**

### 5.2.1 Security audit (FAU)

#### 5.2.1.1 Protected audit trail storage (FAU_STG.1)

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

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

#### 5.2.1.2 Audit review (FAU_SAR.1)

**FAU_SAR.1.1** The ~~TSF~~ **IT environment** shall provide [**the Administrator**] with the capability to read [**all data**] from the audit records.

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

### 5.2.2 Cryptographic support (FCS)

#### 5.2.2.1 Cryptographic key generation (FCS_CKM.1a)

**FCS_CKM.1a.1** The ~~TSF~~ **IT environment** shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**a software random number generator**] and specified cryptographic key sizes [**128 bits**] that meet the following: [**FIPS-197 AES**].

#### 5.2.2.2 Cryptographic key generation (FCS_CKM.1b)

**FCS_CKM.1b.1** The ~~TSF~~ **IT environment** shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**a prime number generator**] and specified cryptographic key sizes [**1024 bits**] that meet the following: [**FIPS-186-2/PKCS #1**].

#### 5.2.2.3 Cryptographic operation (FCS_COP.1)

**FCS_COP.1.1** The ~~TSF~~ **IT environment** shall perform [**public key cryptographic operations**] in accordance with a specified cryptographic algorithm [**RSA**] and cryptographic key sizes [**1024 bits**] that meet the following: [**PKCS#1(RSAES-PKCS1-v1_5)**].

### 5.2.3 Identification and authentication (FIA)

#### 5.2.3.1 User authentication before any action (FIA_UAU.2)

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

### 5.2.4 Security management (FMT)

#### 5.2.4.1 Management of security functions behavior (FMT_MOF.1)

**FMT_MOF.1.1** The ~~TSF~~ **IT environment** shall restrict the ability to [***disable, enable***] the functions [**security audit and decommissioning service**] to [**the Administrator**].

#### 5.2.4.2 Management of security attributes (FMT_MSA.1)

**FMT_MSA.1.1** The ~~TSF~~ **IT environment** shall enforce the [**RMS Use License Access Control Policy**] to restrict the ability to [***modify***] the security attributes [**of the RMS Use License Access Control Policy, including explicit exclusions**] to [**Administrator**].

#### 5.2.4.3 Management of RMS services (FMT_MSA_EX.1)

**FMT_MSA_EX.1.1** The IT environment shall restrict the ability to modify the security attributes controlling access to the decommissioning and sub-enrollment services to the Administrator.

#### 5.2.4.4 Security roles (FMT_SMR.1)

**FMT_SMR.1.1** The ~~TSF~~ **IT environment** shall maintain the roles [**Administrator**].
**FMT_SMR.1.2** The TSF shall be able to associate users with roles.

### 5.2.5 Protection of the TSF (FPT)

#### 5.2.5.1 Protection of RMS services (FPT_RMS_EX.1)

**FPT_RMS_EX.1.1** The IT environment shall limit access to the RMS decommissioning and sub-enrollment services to users authorized by an authorized administrator.

#### 5.2.5.2 Non-bypassability of the TSP (FPT_RVM.1)

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

#### 5.2.5.3 TSF domain separation (FPT_SEP.1)

**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.

#### 5.2.5.4 Reliable time stamps (FPT_STM.1)

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

## 5.3 TOE Security Assurance Requirements

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

| Requirement Class | Requirement Component |
|---|---|
| **ACM: Configuration management** | ACM_AUT.1: Partial CM automation |
| | ACM_CAP.4: Generation support and acceptance procedures |
| | ACM_SCP.2: Problem tracking CM coverage |
| **ADO: Delivery and operation** | ADO_DEL.2: Detection of modification |
| | ADO_IGS.1: Installation, generation, and start-up procedures |
| **ADV: Development** | ADV_FSP.2: Fully defined external interfaces |
| | ADV_HLD.2: Security enforcing high-level design |
| | ADV_IMP.1: Subset of the implementation of the TSF |
| | ADV_LLD.1: Descriptive low-level design |
| | ADV_RCR.1: Informal correspondence demonstration |
| | ADV_SPM.1: Informal TOE security policy model |
| **AGD: Guidance documents** | AGD_ADM.1: Administrator guidance |
| | AGD_USR.1: User guidance |
| **ALC: Life cycle support** | ALC_DVS.1: Identification of security measures |
| | ALC_FLR.3: Systematic flaw remediation |
| | 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 3 EAL 4 augmented with ALC_FLR.3 Assurance Components**

### 5.3.1 Configuration management (ACM)

#### 5.3.1.1 Partial CM automation (ACM_AUT.1)

**ACM_AUT.1.1d** The developer shall use a CM system.
**ACM_AUT.1.2d** The developer shall provide a CM plan.
**ACM_AUT.1.1c** The CM system shall provide an automated means by which only 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 Generation support and acceptance procedures (ACM_CAP.4)

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

### 5.3.1.3 Problem tracking CM coverage (ACM_SCP.2)

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

## 5.3.2 Delivery and operation (ADO)

### 5.3.2.1 Detection of modification (ADO_DEL.2)

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

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

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

**ADO_IGS.1.2e** The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

## 5.3.3 Development (ADV)

### 5.3.3.1 Fully defined external interfaces (ADV_FSP.2)

**ADV_FSP.2.1d** The developer shall provide a functional specification.

**ADV_FSP.2.1c** The functional specification shall describe the TSF and its external interfaces using an informal style.

**ADV_FSP.2.2c** The functional specification shall be internally consistent.

**ADV_FSP.2.3c** The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing complete details of all effects, exceptions and error messages.

**ADV_FSP.2.4c** The functional specification shall completely represent the TSF.

**ADV_FSP.2.5c** The functional specification shall include rationale that the TSF is completely represented.

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

**ADV_FSP.2.2e** The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

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

**ADV_HLD.2.1d** The developer shall provide the high-level design of the TSF.

**ADV_HLD.2.1c** The presentation of the high-level design shall be informal.

**ADV_HLD.2.2c** The high-level design shall be internally consistent.

**ADV_HLD.2.3c** The high-level design shall describe the structure of the TSF in terms of subsystems.

**ADV_HLD.2.4c** The high-level design shall describe the security functionality provided by each subsystem of the TSF.

**ADV_HLD.2.5c** The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

**ADV_HLD.2.6c** The high-level design shall identify all interfaces to the subsystems of the TSF.

**ADV_HLD.2.7c** The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

**ADV_HLD.2.8c** The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.

**ADV_HLD.2.9c** The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.

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

**ADV_HLD.2.2e** The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

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

**ADV_IMP.1.1d** The developer shall provide the implementation representation for a selected subset of the TSF.

**ADV_IMP.1.1c** The implementation representation shall unambiguously define the TSF to a level of detail such that the TSF can be generated without further design decisions.

**ADV_IMP.1.2c** The implementation representation shall be internally consistent.

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

**ADV_IMP.1.2e** The evaluator shall determine that the least abstract TSF representation provided is an accurate and complete instantiation of the TOE security functional requirements.

### 5.3.3.4 Descriptive low-level design (ADV_LLD.1)

**ADV_LLD.1.1d** The developer shall provide the low-level design of the TSF.

**ADV_LLD.1.1c** The presentation of the low-level design shall be informal.

**ADV_LLD.1.2c** The low-level design shall be internally consistent.

**ADV_LLD.1.3c** The low-level design shall describe the TSF in terms of modules.

**ADV_LLD.1.4c** The low-level design shall describe the purpose of each module.

**ADV_LLD.1.5c** The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.

**ADV_LLD.1.6c** The low-level design shall describe how each TSP-enforcing function is provided.

**ADV_LLD.1.7c** The low-level design shall identify all interfaces to the modules of the TSF.

**ADV_LLD.1.8c** The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.

**ADV_LLD.1.9c** The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing details of effects, exceptions and error messages, as appropriate.

**ADV_LLD.1.10c**The low-level design shall describe the separation of the TOE into TSP-enforcing and other modules.

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

**ADV_LLD.1.2e** The evaluator shall determine that the low-level design is an accurate and complete instantiation of the TOE security functional requirements.

#### 5.3.3.5 Informal correspondence demonstration (ADV_RCR.1)

**ADV_RCR.1.1d** The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.

**ADV_RCR.1.1c** For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

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

#### 5.3.3.6 Informal TOE security policy model (ADV_SPM.1)

**ADV_SPM.1.1d** The developer shall provide a TSP model.

**ADV_SPM.1.2d** The developer shall demonstrate correspondence between the functional specification and the TSP model.

**ADV_SPM.1.1c** The TSP model shall be informal.

**ADV_SPM.1.2c** The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modeled.

**ADV_SPM.1.3c** The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modeled.

**ADV_SPM.1.4c** The demonstration of correspondence between the TSP model and the functional specification shall show that all of the security functions in the functional specification are consistent and complete with respect to the TSP model.

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

### 5.3.4 Guidance documents (AGD)

#### 5.3.4.1 Administrator guidance (AGD_ADM.1)

**AGD_ADM.1.1d**The developer shall provide administrator guidance addressed to system administrative personnel.

**AGD_ADM.1.1c**The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.

**AGD_ADM.1.2c**The administrator guidance shall describe how to administer the TOE in a secure manner.

**AGD_ADM.1.3c**The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

**AGD_ADM.1.4c**The administrator guidance shall describe all assumptions regarding user 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 User guidance (AGD_USR.1)

**AGD_USR.1.1d** The developer shall provide user guidance.

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

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

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

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

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

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

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

## 5.3.5 Life cycle support (ALC)

### 5.3.5.1 Identification of security measures (ALC_DVS.1)

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

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

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

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

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

### 5.3.5.2 Systematic flaw remediation (ALC_FLR.3)

**ALC_FLR.3.1d** The developer shall provide flaw remediation procedures addressed to TOE developers.

**ALC_FLR.3.2d** 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.3.3d** The developer shall provide flaw remediation guidance addressed to TOE users.

**ALC_FLR.3.1c** The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

**ALC_FLR.3.2c** 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.3.3c** The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

**ALC_FLR.3.4c** 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.3.5c** 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.3.6c** The procedures for processing reported security flaws shall ensure that any reported flaws are corrected and the correction issued to TOE users.

**ALC_FLR.3.7c** 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.3.8c** 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.3.9c** The flaw remediation procedures shall include a procedure requiring timely responses for the automatic distribution of security flaw reports and the associated corrections to registered users who might be affected by the security flaw.

**ALC_FLR.3.10c** The flaw remediation guidance shall describe a means by which TOE users may register with the developer, to be eligible to receive security flaw reports and corrections.

**ALC_FLR.3.11c** The flaw remediation guidance shall identify the specific points of contact for all reports and enquiries about security issues involving the TOE.

**ALC_FLR.3.1e** The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

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

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

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

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

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

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

#### 5.3.5.4 Well-defined development tools (ALC_TAT.1)

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

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

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

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

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

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

### 5.3.6 Tests (ATE)

#### 5.3.6.1 Analysis of coverage (ATE_COV.2)

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

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

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

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

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

**ATE_DPT.1.1d** The developer shall provide the analysis of the depth of testing.
**ATE_DPT.1.1c** The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design.
**ATE_DPT.1.1e** The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

#### 5.3.6.3 Functional testing (ATE_FUN.1)

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

#### 5.3.6.4 Independent testing - sample (ATE_IND.2)

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

### 5.3.7 Vulnerability assessment (AVA)

#### 5.3.7.1 Validation of analysis (AVA_MSU.2)

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

**AVA_MSU.2.3e** The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.

**AVA_MSU.2.4e** The evaluator shall confirm that the analysis documentation shows that guidance is provided for secure operation in all modes of operation of the TOE.

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

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

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

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

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

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

### 5.3.7.3 Independent vulnerability analysis (AVA_VLA.2)

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

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

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

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

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

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

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

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

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

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

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

# 6. TOE Summary Specification

This chapter describes the security functions and associated assurance measures.

## 6.1 TOE Security Functions

The TOE implements the following security functions:

- Security audit
- User data protection,
- Identification and authentication, and
- Security management.

### 6.1.1 Security audit

Windows RMS installs support for logging during the initial RMS system setup; this logging service is enabled and started automatically. An RMS logging database is created in the SQL server database instance that is also used for the configuration database. Note that the SQL database server is a component of the TOE environment.

Administrators (in the TOE environment) can enable or disable the logging service at any time. When enabled, the logging service will send all data about RMS requests to the logging database. Administrators can write SQL scripts to pare down the information so that only the specific information required by the organization is stored. RMS logs both successful and failed use license requests. This log enables an organization to record both the users who have successfully accessed rights-protected information and the users who attempted unauthorized access of rights-protected information.

The report logs stored in the SQL database contain the following information:

- HostMachineName - The computer that handled the request.
- HostMachineRequestId - Uniquely identifies this request on this computer.
- RequestTime - Time, in Coordinated Universal Time (Greenwich Mean Time), when the request was received.
- RequestPath - Relative URL to the .asmx file, for example: /_wmcs/licensing/License.asmx.
- RequestType - Name of the Web method invoked, for example: AcquireLicense.
- RequestUserAddress - Requestor’s source IP address.
- RequestUserAgent - HTTP header User Agent value.
- AuthenticatedState - Whether the HTTP connection was authenticated (True/False).
- SecureConnectionState - Whether or not it is an SSL connection (True/False).
- AuthenticatedId - Logon name for authenticated requests. Blank if AuthenticatedState=False.
- ReceivedXrML - The XrML document that is received from the requestor.
- IssuedXrML - The XrML document returned to the requestor.
- SuccessOrFailure - Whether the request succeeded or failed (Succeeded/Failed).
- Metadata - Metadata field.
- ErrorInformation - Descriptive error message, if an error occurred.

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

- FAU_GEN_EX.1: All use license requests are recorded in the report log along with contents summarized above.

### 6.1.2 User data protection

**License Generation**

License requests for use licenses that are made by an RMS client must include a rights account certificate in a request. Publishing licenses contain usage rights for encrypted data that identifies the users who can view the content, as well as in what manner they can edit and distribute the content. Publishing licenses contain the symmetric content key for decrypting the content that is encrypted with the public key of the RMS server, to ensure that only the server can decrypt the content key and issue use licenses. Use licenses contain the symmetric content key for decrypting the content and it is encrypted with the public key of the user's rights account certificate, to ensure that only the requesting user can consume the RMS-protected content. Each user who receives RMS-protected content and who is named in the publishing license for that content can request a use license. Both publishing licenses and use licenses are signed by the TOE.

**Certificate Generation**

Certificate requests for rights account certificates made by an RMS client must be first authenticated to the operating system in the IT environment. Rights account certificates associate user accounts with specific computers. Rights account certificates are signed by the TOE.

**Certificate Contents**

At a minimum, the TSF shall ensure that:

If the TSF generates rights account certificates, at a minimum, the TSF shall ensure that:

1. A user is identified.
2. The user's public key is included.
3. A specific computer is identified.
4. A validity period is identified.
5. The identity of the instance of the TOE that issued the certificate is identified.

If the TSF generates server licensor certificates, at a minimum, the TSF shall ensure that:

1. A server is identified.
2. The server's corresponding public key is included.
3. A validity period is identified.
4. The identity of the instance of the TOE that issued the certificate is identified.

If the TSF generates client licensor certificates, at a minimum, the TSF shall ensure that:

1. A user is identified.
2. The user's corresponding public key is included.
3. The user's corresponding private key is included.
4. The user's corresponding private key is encrypted using the rights account certificate public key of the client licensor requestor.
5. A validity period is identified.
6. The identity of the instance of the TOE that issued the certificate is identified.

**License Contents**

If the TSF generates publishing licenses, at a minimum, the TSF shall ensure that:

1. The identity of the content that the license is granting rights to.
2. The rights that are granted to the content.
3. The identity of one or more users that the license is granting rights to are identified.
4. The identity of the instance of the TOE that issued the license is identified.
5. A validity period is identified.

If the TSF generates use licenses in 'online' publishing mode, at a minimum, the TSF shall ensure that:

1. A user is identified.
2. The symmetric content key for decrypting the content that is encrypted with the user's public key.
3. The identity of the instance of the TOE that issued the use license is identified.
4. A validity period is identified.

For both certificates and licenses, the TOE implements a trust hierarchy. At each level of the certificate chain, the TOE validates the license or certificate, and then verifies that it connects to a known root of trust through a chain of trust. Each license or certificate that is in the chain is checked by the TSF to validate the following conditions:

1. The issuer signature is valid.
2. The actual use of the license or certificate matches the intended use.
3. The validity dates are met.
4. The license signature key and certified issuer key match.

Beyond certificate and license management, the TOE also enforces an access control policy for the issuance of use licenses for RMS-protected content objects. In order to obtain a use license the following conditions need to be satisfied:

1. the requesting user's account certificate must be valid;
2. the publishing license associated with the RMS-protected content object must be valid;
3. the requesting user or one of its groups must be identified in the publishing license; and
4. the TSF must be able to successfully decrypt the applicable symmetric content key using the server's private key.

Furthermore, if the user has been explicitly excluded from obtaining use licenses by an administrator, the user cannot obtain a use license regardless of the conditions above. Similarly, a server licensor certificate can only be obtained if the sub-enrollment services can be accessed (note that access to this service is controlled by the IT environment). Alternately, if the decommissioning service is enabled none of the other services can be also enabled and the conditions above cannot be checked. As such, if the user can contact the decommissioning service, they would have complete access to any applicable RMS-protected content objects.

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

- FDP_ACC.1 & FDP_ACF.1: The TOE enforces an access policy controlling whether users can obtain use licenses for specific RMS-protected content objects.
- FDP_RMS_EX.1: The TSF ensures appropriate contents when issuing certificates (as identified above) and imposes appropriate restrictions when issuing certificates.
- FDP_RMS_EX.2: The TSF ensures appropriate contents when issuing licenses (as identified above) and imposes appropriate restrictions when issuing licenses.

### 6.1.3 Identification and authentication

The TOE is responsible only for identification. It is expected that identities will be the same as those authenticated in the underlying Windows XP/Server 2003 IT environment component. As such, the TOE obtains already authenticated user identities from the operating system. The TOE will not offer any service with respect to use license requests or client licensor certificate requests without first determining the applicable user identity.

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

- FIA_UID.1: In the TSF, user identities are assumed to be authenticated by the underlying operating system, but must be known to the TSF before any service having to do with use license requests or client licensor certificate requests can be used.

### 6.1.4 Security management

Before the TOE can issue any certificates and licenses, an administrator must define the corresponding validity periods that will be incorporated in the content to be issued. The configuration of these validity periods is accomplished using functions offered by the TOE.

The TOE can be configured to enable the decommissioning service. When that service is enabled, all other services become disabled. Once enabled the decommissioning service cannot be disabled.

The TOE can also be configured to generate audit logs and forward them to the SQL server RMS logging database in the TOE environment. Note that this function is enabled by default when RMS is installed, but can subsequently be disabled and re-enabled as required by the administrator.

All RMS-protected content is protected by default since no use license is ever issued before a user is specifically granted some rights to the corresponding RMS-protected content object. The TOE provides functions to manage the security attributes used in making decisions to issue use licenses, including the ability to define exclusions that would deny specific users the ability to obtain a use license.

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

- FMT_MOF_EX.1: The TSF requires an administrator to define validity periods for issued certificates and licenses.
- FMT_MSA.3: The TSF ensures that no use license is issued to access RMS protected content unless an appropriately identified user is specifically granted usage rights for that RMS protected content.
- FMT_SMF.1: The TSF offers functions to enable and disable the security audit function, to enable the decommissioning service, to manage security attributes used for use license access control decisions, and to manage the templates for certificates and licenses.

## 6.2 TOE Security Assurance Measures

The following security measures correspond directly with the assurance classes of EAL4 augmented with ALC_FLR.3.

### 6.2.1 Configuration management

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

These activities are documented in:

- Windows 2003/XP CM Manual

The Configuration management assurance measure satisfies the following EAL 4 augmented with ALC_FLR.3 assurance requirements:

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

### 6.2.2 Delivery and operation

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

These activities are documented in:

- Microsoft Windows Rights Management Services (RMS) Security Configuration Guide.
- Windows 2003/XP Delivery and Operation Procedures

The Delivery and operation assurance measure satisfies the following EAL 4 augmented with ALC_FLR.3 assurance requirements:

- ADO_DEL.2
- ADO_IGS.1

### 6.2.3 Development

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

These activities are documented in a large number of related documents. These documents are:

- Introduction: Describes the form, content, and organization of the System Design documentation.
- Security Policy: Provides an informal description and model of the access control policy for the system.
- System Decomposition Summary: This document describes the decomposition of the system and identifies the subsystems in terms of components.
- Component Descriptions (several): There are several of these documents; one each for the system components defined in the Decomposition Summary document. Each document describes the component and identifies the modules within the component in terms of subcomponents.
- Subcomponent Designs (many): There are many of these documents; one each for the subcomponents defined in the several Component Description documents. Each subcomponent design document presents the following:
  - Summary identifying the subcomponent's name, implementation location, and execution environment.

  - A description of the design of the subcomponent and a summary of its security functions and mechanisms.
  - A specification of each TSF interface implemented by the subcomponent. The following is provided for each TSF interface: purpose, parameters, security checks, and security effects.
  - A correspondence matrix that identifies for each TSF interface, which security functions the interface's checks and effects help implement. The matrix includes a rationale for this correspondence.
- A test family summary that describes test cases implemented in the security tests for each API.

The Development assurance measure satisfies the following EAL 4 augmented with ALC_FLR.3 assurance requirements:

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

### 6.2.4 Guidance documents

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

These activities are documented in:

- Microsoft Windows Rights Management Services (RMS) Evaluated Configuration Administrator's Guide.
- Microsoft Windows Rights Management Services (RMS) Evaluated Configuration User's Guide.

The Guidance documents assurance measure satisfies the following EAL 4 augmented with ALC_FLR.3 assurance requirements:

- AGD_ADM.1
- AGD_USR.1

### 6.2.5 Life cycle support

Microsoft ensures the adequacy of the procedures used during the development and maintenance of the TOE through the use of a comprehensive life-cycle management plan. Microsoft applies security controls on the development environment that are adequate to provide the confidentiality and integrity of the TOE design and implementation that is necessary to ensure the secure development of the TOE. Microsoft has procedures that define the process for accepting and acting upon user reports of security flaws. These procedures describe the acceptance criteria for security flaws, how all security flaws and the status of fixes for each security flaw are tracked, and how corrections and corrective measures are automatically made available as applicable. Microsoft has a documented model of the TOE life cycle that ensures that the TOE is developed and maintained in a well-defined manner. Microsoftuses well-defined development tools in order to ensure consistent and predictable results while developing the TOE.

These activities are documented in:

- Windows 2003/XP Life Cycle Management Plan

The Life cycle support assurance measure satisfies the following EAL 4 augmented with ALC_FLR.3 assurance requirements:

- ALC_DVS.1
- ALC_FLR.3
- ALC_LCD.1
- ALC_TAT.1

### 6.2.6 Tests

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

These activities are documented in four parts: a *test plan* (“Windows Rights Management Services Security Test Plan”), *test families*, *test suites*, and *test results*.

- The test plan describes the form, content, and organization of test documentation. It also summarizes each of the test suites and includes high-level procedures for exercising the tests.
- The test families described the set of security-relevant test cases on a per-subcomponent basis. These descriptions include references to the corresponding test suites that implement those test cases. Note that every test case corresponds to at least one test suite.
- The test suites include both documentation and an actual implemented test (if applicable). Test suites are organized around tests that share a common theme, such as handle enforcement, privilege enforcement, auditing, etc. The test suite documentation describes the purpose and “theme” for the test suite, the set of test variations that are exercised for each of its corresponding test cases, procedures to successfully exercise the test suite, and the expected results. The test suite documentation also implicitly includes the actual tests that provide specific details regarding test variations and expected results.
- The test results are essentially the set of logs resulting from completely exercising all of the security test procedures. These logs include summaries of the results in terms of total test variations, counts of variations that passed, failed, or blocked (i.e., were unable to run), and detailed information about each variation that was attempted, including more detailed results and expected results.

The Tests assurance measure satisfies the following EAL 4 augmented with ALC_FLR.3 assurance requirements:

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

### 6.2.7 Vulnerability assessment

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

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

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

These activities are documented in:

- The Windows Rights Management Services Vulnerability Analysis

The Vulnerability assessment assurance measure satisfies the following EAL 4 augmented with ALC_FLR.3 assurance requirements:

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

# 7. Protection Profile Claims

There are no PP claims in this ST.

# 8. Rationale

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

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

## 8.1 Security Objectives Rationale

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

### 8.1.1 Security Objectives Rationale for the TOE and Environment

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

| | **T.ACCESS** | **T.CRYPTO** | **T.IMPERSONATE** | **T.MANAGE** | **T.PROTECT** | **T.UNDETECTED_ACTIONS** | **A.COOP** | **A.LOCATE** | **A.MANAGE** | **A.NO_EVIL_ADM** | **A.PEER** | **A.PROTECT** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **O.CONTENT** | X | | | | | | | | | | | |
| **O.IDENTIFICATION** | | | X | | | | | | | | | |
| **O.MANAGE** | | | | X | | | | | | | | |
| **O.REPORTING** | | | | | | X | | | | | | |
| **O.ADMIN** | | | | X | | | | | | | | |
| **O.AUDITSTORE** | | | | | | X | | | | | | |
| **O.AUTHENTICATION** | | | X | | | | | | | | | |
| **O.CRYPTO** | | X | | | | | | | | | | |
| **O.PROTECT** | | | | | X | | | | | | | |
| **O.CREDEN** | | | | | | | X | | | | | |
| **O.INSTALL** | | | | | | | | | X | X | X | |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **O.PHYSICAL** | | | | | | | | X | | | | X |
| **O.TIME** | | | | | | X | | | | | | |

**Table 4 Environment to Objective Correspondence**

#### 8.1.1.1 T.ACCESS

*The TOE may provide information inappropriately or fail to provide the correct information in response to a valid request.*

This Threat is satisfied by ensuring that:

- O.CONTENT: The TOE will ensure that certificates and licenses have the appropriate contents when issued and that they are only issued when it is appropriate to do so.

#### 8.1.1.2 T.CRYPTO

*The TOE may fail to use appropriate cryptographic services to support its cryptographic operations.*

This Threat is satisfied by ensuring that:

- O.CRYPTO: The IT environment will provide cryptographic functions to support the TOE’s cryptographic operations.

#### 8.1.1.3 T.IMPERSONATE

*The TOE may fail to properly identify a user making a request for service.*

This Threat is satisfied by ensuring that:

- O.IDENTIFICATION: The TOE will identify each user before allowing its services to be accessed.
- O.AUTHENTICATION: The IT Environment will authenticate users before allowing them to communicate with the TOE.

#### 8.1.1.4 T.MANAGE

*The TOE may fail to provide the functions necessary to effectively manage its own security functions.*

This Threat is satisfied by ensuring that:

- O.MANAGE: The TOE will provide the set of functions necessary to effectively manage the TOE security functions.
- O.ADMIN: The IT environment will implement an administrator role for the TOE.

#### 8.1.1.5 T.PROTECT

*The TOE may be bypassable or subject to tampering by untrusted users.*

This Threat is satisfied by ensuring that:

- O.PROTECT: The IT environment will ensure that the TOE is protected from bypass and tampering attempts.

#### 8.1.1.6 T.UNDETECTED_ACTIONS

*An unauthorized user may perform unauthorized actions that go undetected because of the failure of the system to record actions.*

This Threat is satisfied by ensuring that:

- O.REPORTING: The TOE records use license requests.
- O.AUDITSTORE: The IT environment provides the means of storing and retrieving generated audit logs.

- O.TIME: The IT environment supports this objective by ensuring reliable time information is available for the audit logs.

#### 8.1.1.7 A.COOP

*Authorized users possess the necessary authorization to access at least some of the information managed by the TOE and are expected to act in a cooperating manner in a benign environment.*

This Assumption is satisfied by ensuring that:

- O.CREDEN: Those responsible for the TOE will ensure that all access credentials are protected by the applicable users.

#### 8.1.1.8 A.LOCATE

*The processing resources of the TOE will be located within controlled access facilities which will prevent unauthorized physical access.*

This Assumption is satisfied by ensuring that:

- O.PHYSICAL: Those responsible for the TOE will ensure that the TOE is appropriately protected from physical attacks.

#### 8.1.1.9 A.MANAGE

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

This Assumption is satisfied by ensuring that:

- O.INSTALL: Those responsible for the TOE will operate the TOE in a manner that ensures the integrity of its security features.

#### 8.1.1.10 A.NO_EVIL_ADM

*The system administrative personnel are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the administrator documentation.*

This Assumption is satisfied by ensuring that:

- O.INSTALL: Those responsible for the TOE will operate the TOE in a manner that ensures the integrity of its security features.

#### 8.1.1.11 A.PEER

*Any other systems with which the TOE communicates are assumed to be under the same management control and operate under the same security policy constraints.*

This Assumption is satisfied by ensuring that:

- O.INSTALL: Those responsible for the TOE will ensure that the TOE is properly installed and configured.

#### 8.1.1.12 A.PROTECT

*The TOE hardware and software critical to security policy enforcement will be protected from unauthorized physical modification.*

This Assumption is satisfied by ensuring that:

- O.PHYSICAL: Those responsible for the TOE will ensure that the TOE is appropriately protected from physical attacks.

## 8.2 Security Requirements Rationale

This section provides evidence supporting the internal consistency and completeness of the components (requirements) in the Security Target. Note that **Table 5** indicates the requirements that effectively satisfy the individual objectives. .

### 8.2.1 Security Functional Requirements Rationale

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

| | **O.CONTENT** | **O.IDENTIFICATION** | **O.MANAGE** | **O.REPORTING** | **O.ADMIN** | **O.AUDITSTORE** | **O.AUTHENTICATION** | **O.CRYPTO** | **O.PROTECT** | **O.TIME** |
|---|---|---|---|---|---|---|---|---|---|---|
| **FAU_GEN_EX.1** | | | | X | | | | | | |
| **FDP_ACC.1** | X | | | | | | | | | |
| **FDP_ACF.1** | X | | | | | | | | | |
| **FDP_RMS_EX.1** | X | | | | | | | | | |
| **FDP_RMS_EX.2** | X | | | | | | | | | |
| **FIA_UID.1** | | X | | | | | | | | |
| **FMT_MOF_EX.1** | X | | X | | | | | | | |
| **FMT_MSA.3** | | | X | | | | | | | |
| **FMT_SMF.1** | | | X | | | | | | | |
| **FAU_STG.1** | | | | | | X | | | | |
| **FAU_SAR.1** | | | | | | X | | | | |
| **FCS_CKM.1a** | | | | | | | | X | | |
| **FCS_CKM.1b** | | | | | | | | X | | |
| **FCS_COP.1** | | | | | | | | X | | |
| **FIA_UAU.2** | | | | | | | X | | | |
| **FMT_MOF.1** | | | | X | | | | | X | |
| **FMT_MSA.1** | | | | | | | | | X | |
| **FMT_MSA_EX.1** | | | | | | | | | X | |
| **FMT_SMR.1** | | | | | X | | | | | |
| **FPT_RMS_EX.1** | | | | | | | | | X | |
| **FPT_RVM.1** | | | | | | | | | X | |
| **FPT_SEP.1** | | | | | | | | | X | |
| **FPT_STM.1** | | | | | | | | | | X |

**Table 5 Objective to Requirement Correspondence**

#### 8.2.1.1 O.CONTENT

*The TOE must ensure that certificates and licenses have appropriate content when issued to users and are only issued when appropriate.*

This TOE Security Objective is satisfied by ensuring that:

- FDP_ACC.1 and FDP_ACF.1: The TSF will ensure that users are granted use licenses only when they are authorized some access to the applicable RMS-protected content object.
- FDP_RMS_EX.2: The TSF will ensure that certificates have the appropriate contents when issued.
- FDP_RMS_EX.3: The TSF will ensure that licenses have the appropriate contents when issued.
- FMT_MOF_EX.1: The TSF will require administrators to define certificate and license validity periods.

#### 8.2.1.2 O.IDENTIFICATION

*The TOE must be designed to ensure that it can identify each user before providing client licensor certificates or use licenses.*

This TOE Security Objective is satisfied by ensuring that:

- FIA_UID.1: The TSF will ensure that users are identified before providing client licensor certificates or use licenses.

#### 8.2.1.3 O.MANAGE

*The TOE must provide administrator functions suitable to manage the security functions of the TOE.*

This TOE Security Objective is satisfied by ensuring that:

- FMT_MOF_EX.1: The TSF will require administrators to define certificate and license validity periods.
- FMT_MSA.3: The TSF will ensure that RMS-protected content is protected by default.
- FMT_SMF.1: The TSF will provide the function necessary to manage the security audit and use license access control functions and decommissioning service.

#### 8.2.1.4 O.REPORTING

*The TSF must record use license requests.*

This TOE Security Objective is satisfied by ensuring that:

- FAU_GEN_EX.1: The TOE will generate records of use license requests.
- FMT_MOF.1: The IT environment restricts the ability to enable and disable the audit function to the administrator.

#### 8.2.1.5 O.ADMIN

*The IT environment is responsible to ensure that an administrator role is established for the management of the TOE.*

This IT Environment Security Objective is satisfied by ensuring that:

- FMT_SMR.1: The IT environment will ensure that there is an administrator role for the purpose of managing the TOE.

#### 8.2.1.6 O.AUDITSTORE

*The IT Environment is responsible to store and make accessible audit records generated by the TOE.*

This IT Environment Security Objective is satisfied by ensuring that:

- FAU_STG.1: The IT environment will store and protect audit logs generated by the TOE.
- FAU_SAR.1: The IT environment will make stored audit records accessible.

#### 8.2.1.7 O.AUTHENTICATION

*The IT Environment is responsible to authenticate users before allowing their requests to be passed to the TOE.*

This IT Environment Security Objective is satisfied by ensuring that:

- FIA_UAU.2: The IT environment will ensure that users are identified before offering its services.

#### 8.2.1.8 O.CRYPTO

*The IT environment is responsible to provide appropriate cryptographic functions in support of the cryptographic operations of the TOE.*

This IT Environment Security Objective is satisfied by ensuring that:

- FCS_CKM.1a: The IT environment will provide the ability to generate symmetric keys.
- FCS_CKM.1b: The IT environment will provide the ability to generate asymmetric keys.
- FCS_COP.1: The IT environment will provide the ability to perform public-key cryptographic operations (e.g., encryption, signing, signature verification).

#### 8.2.1.9 O.PROTECT

*The IT environment is responsible to ensure that the TOE is not bypassed and is not subject to tampering.*

This IT Environment Security Objective is satisfied by ensuring that:

- FPT_RMS_EX.1: The IT environment will appropriately control access to applicable RMS services.
- FPT_RVM.1: The IT environment will protect the TOE such that its security functions cannot be bypassed.
- FPT_SEP.1: The IT environment will protect the TOE from potential tampering attempts.
- FMT_MOF.1: The IT environment will ensure that security function operation is controllable only by the administrator.
- FMT_MSA.1: The IT environment will ensure that access to RMS-protected content is controllable only by the administrator.
- FMT_MSA_EX.1: The IT environment will restrict the ability to manage access to applicable RMS services to the administrator.

#### 8.2.1.10 O.TIME

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

This IT Environment Security Objective is satisfied by ensuring that:

- FPT_STM.1: The IT environment will provide a timestamp to the TOE for use in reporting records.

## 8.3 Security Assurance Requirements Rationale

This ST contains the assurance requirements from the CC EAL4 assurance package augmented with ALC_FLR.3. The CC allows assurance packages to be augmented, which allows the addition of assurance components from the CC not already included in the EAL. Augmentation was chosen to provide the added assurance that is provided by defining flaw remediation procedures and correcting security flaws (ALC_FLR.3). This ST is based on good rigorous commercial development practices and has been developed for a generalized environment for a TOE that is generally available and does not require modification to meet the security needs of the environment specified in this ST.

## 8.4 Strength of Functions Rationale

The TOE does not include any probabilistic or permutational mechanisms, except those related to some of its cryptographic operations. As such, while SOF-medium should be considered consistent with EAL 4 augmented with ALC_FLR.3 it is not applicable to this TOE.

## 8.5 Requirement Dependency Rationale

The following table identifies the dependencies of the requirements in this ST, including the requirements explicitly defined in this ST. As indicated in the table, all of the dependencies are satisfied, except those for FCS_COP.1, FCS_CKM.1a, and FCS_CKM.1b. Those requirements have been identified for the environment to support the functions of the TOE. It is expected that the environment would further satisfy its own requirements, and the TOE is not dependent upon the manner in which that is accomplished. As such, the applicable dependencies are not specifically identified in this ST. Note also that the intended environment of the TOE has FIPS-certified cryptographic functions which should serve to indicate that any dependencies are actually addressed.

Note that the IT environment SFRs are identified using *italics*, the assurance requirements are underlined, and the missing dependencies are identified in [**bold-red-bracketed]** text.

| ST Requirement | CC Dependencies | ST Dependencies |
|---|---|---|
| **FAU_GEN_EX.1** | FPT_STM.1 | **FPT_STM.1** |
| **FDP_ACC.1** | FDP_ACF.1 | FDP_ACF.1 |
| **FDP_ACF.1** | FDP_ACC.1 and FMT_MSA.3 | FDP_ACC.1 and FMT_MSA.3 |
| **FDP_RMS_EX.1** | none | none |
| **FDP_RMS_EX.2** | none | none |
| **FIA_UID.1** | none | none |
| **FMT_MOF_EX.1** | FMT_SMR.1 | *FMT_SMR.1* |
| **FMT_MSA.3** | FMT_MSA.1 and FMT_SMR.1 | *FMT_MSA.1* and *FMT_SMR.1* |
| **FMT_SMF.1** | none | none |
| **FAU_STG.1** | FAU_GEN.1 | FAU_GEN.1 |
| **FAU_SAR.1** | FAU_GEN.1 | FAU_GEN.1 |
| **FCS_CKM.1a** | FCS_CKM.2 or FCS_COP.1) and FCS_CKM.4 and FMT_MSA.2 | *FCS_COP.1* and **[FCS_CKM.4]** and **[FMT_MSA.2]** |
| **FCS_CKM.1b** | FCS_CKM.2 or FCS_COP.1) and FCS_CKM.4 and FMT_MSA.2 | *FCS_COP.1* and **[FCS_CKM.4]** and **[FMT_MSA.2]** |
| **FCS_COP.1** | FDP_ITC.1 or FCS_CKM.1) and FCS_CKM.4 and FMT_MSA.2 | FCS_CKM.1b and **[FCS_CKM.4]** and **[FMT_MSA.2]** |
| **FIA_UAU.2** | FIA_UID.1 | FIA_UID.1 |
| **FMT_MOF.1** | FMT_SMR.1 and FMT_SMF.1 | *FMT_SMR.1* and FMT_SMF.1 |
| **FMT_MSA.1** | FMT_SMR.1 and FMT_SMF.1 and (FDP_ACC.1 or FDP_IFC.1) | *FMT_SMR.1* and FMT_SMF.1 and FDP_ACC.1 |
| **FMT_MSA_EX.1** | FMT_SMR.1 | *FMT_SMR.1* |
| **FMT_SMR.1** | FIA_UID.1 | FIA_UID.1 |
| **FPT_RMS_EX.1** | none | none |
| **FPT_RVM.1** | none | none |
| **FPT_SEP.1** | none | none |
| **FPT_STM.1** | none | none |
| **ACM_AUT.1** | ACM_CAP.3 | ACM_CAP.4 |
| **ACM_CAP.4** | ALC_DVS.1 | ALC_DVS.1 |
| **ACM_SCP.2** | ACM_CAP.3 | ACM_CAP.4 |
| **ADO_DEL.2** | ACM_CAP.3 | ACM_CAP.4 |
| **ADO_IGS.1** | AGD_ADM.1 | AGD_ADM.1 |
| **ADV_FSP.2** | ADV_RCR.1 | ADV_RCR.1 |
| **ADV_HLD.2** | ADV_FSP.1 and ADV_RCR.1 | ADV_FSP.2 and ADV_RCR.1 |
| **ADV_IMP.1** | ADV_LLD.1 and ADV_RCR.1 and ALC_TAT.1 | ADV_LLD.1 and ADV_RCR.1 and ALC_TAT.1 |
| **ADV_LLD.1** | ADV_HLD.2 and ADV_RCR.1 | ADV_HLD.2 and ADV_RCR.1 |
| **ADV_RCR.1** | none | none |
| **ADV_SPM.1** | ADV_FSP.1 | ADV_FSP.2 |
| **AGD_ADM.1** | ADV_FSP.1 | ADV_FSP.2 |
| **AGD_USR.1** | ADV_FSP.1 | ADV_FSP.2 |
| **ALC_DVS.1** | none | none |
| **ALC_FLR.3** | none | none |
| **ALC_LCD.1** | none | none |
| **ALC_TAT.1** | ADV_IMP.1 | ADV_IMP.1 |
| **ATE_COV.2** | ADV_FSP.1 and ATE_FUN.1 | ADV_FSP.2 and ATE_FUN.1 |

| | | |
|---|---|---|
| **ATE_DPT.1** | ADV_HLD.1 and ATE_FUN.1 | ADV_HLD.2 and ATE_FUN.1 |
| **ATE_FUN.1** | none | none |
| **ATE_IND.2** | ADV_FSP.1 and AGD_ADM.1 and AGD_USR.1 and ATE_FUN.1 | ADV_FSP.2 and AGD_ADM.1 and AGD_USR.1 and ATE_FUN.1 |
| **AVA_MSU.2** | ADO_IGS.1 and ADV_FSP.1 and AGD_ADM.1 and AGD_USR.1 | ADO_IGS.1 and ADV_FSP.2 and AGD_ADM.1 and AGD_USR.1 |
| **AVA_SOF.1** | ADV_FSP.1 and ADV_HLD.1 | ADV_FSP.2 and ADV_HLD.2 |
| **AVA_VLA.2** | ADV_FSP.1 and ADV_HLD.2 and ADV_IMP.1 and ADV_LLD.1 and AGD_ADM.1 and AGD_USR.1 | ADV_FSP.2 and ADV_HLD.2 and ADV_IMP.1 and ADV_LLD.1 and AGD_ADM.1 and AGD_USR.1 |

**Table 6 Dependency Analysis**

## 8.6 Explicitly Stated Requirements Rationale

This ST includes a number of explicitly stated security functional requirements. Those requirements are organized in two new families of requirements within existing functional classes as well as four new components within existing functional families.

- **Security Audit** – One new component **(**FAU_GEN_EX.1) has been introduced to the FAU_GEN family of requirements and are modeled after CC audit requirements. It has been explicitly stated because the information written to the logs is not conformant with the CC required audit events.
- **User Data Protection** - The Content License management family of security functional requirements has been developed to introduce requirements to ensure that certificates, licenses, and certificate and license status information is exported and has the appropriate contents. There are two components within this family: the first (FDP_RMS_EX.1) addresses the content requirements for certificates and the second (FDP_RMS_EX.2) addresses the content requirements for licenses. Note that none of these components has any dependency nor are they organized hierarchically.
- **Security management** - Two new components (FMT_MOF_EX.1 and FMT_MSA_EX.1) have been introduced to the FMT_MOF and FMT_MSA families of requirements. FMT_MOF_EX.1 has been added to address the need to require administrator-defined validity periods in certificates and licenses. Note that this component is dependent upon FMT_SMR.1. FMT_MSA_EX.1 has been added to address the need to control the ability to control access to RMS services in the environment. It is dependent on FMT_SMR.1 given reference to the administrator role.

Note that the CC does not contain requirements that are readily adaptable to these specific needs.

## 8.7 TOE Summary Specification Rationale

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

This Section in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions are suitable to meet the TOE security requirements. The collection of security functions work together to provide all of the security requirements. The security functions described in the TOE summary specification are all necessary for the required security functionality in the TSF. **Table 7 Security Functions vs. Requirements Mapping** demonstrates the relationship between security requirements and security functions.

| | **Security Audit** | **User data protection** | **Identification and authentication** | **Security management** |
|---|---|---|---|---|
| **FAU_GEN_EX.1** | X | | | |
| **FDP_ACC.1** | | X | | |
| **FDP_ACF.1** | | X | | |
| **FDP_RMS_EX.1** | | X | | |
| **FDP_RMS_EX.2** | | X | | |
| **FIA_UID.1** | | | X | |
| **FMT_MOF_EX.1** | | | | X |
| **FMT_MSA.3** | | | | X |
| **FMT_SMF.1** | | | | X |

**Table 7 Security Functions vs. Requirements Mapping**

## 8.8 PP Claims Rationale

See Section 7, Protection Profile Claims.