CloudMask

CloudMask Engine v2.0

Security Target

Oct 2015

Document prepared by

Document History

Version	Date	Author	Description
0.1	11 June 2014	A Boulton	Initial draft.
0.2	18 June 2014	A Boulton	Initial Vendor comments and corrections.
0.3	20 June 2014	A Boulton	Completion of SFRs
0.4	07 July 2014	A Boulton	Rationale section updated, SFRs revised, changes to SPD.
0.5	16 July 2014	A Boulton	Comments from internal review addressed, consistency checking.
1.0	17 July 2014	A Boulton	Evaluation deliverable for ASE
1.1	08 August 2014	A Boulton	Response to OR1, removed crypto SFR
1.2	22 August 2014	A Boulton	Address remaining open items from OR1
1.3	12 September	A Boulton	Address Certifier OR
1.4	06 October 2014	A Boulton	Address Certifier OR and discussion comments
1.5	14 October 2014	A Boulton	Address evaluator comments
1.6	20 October	A Boulton	Internal revisions
1.7	07 November 2014	A Boulton	Address Certifier OR2
1.8	06 May 2015	TG	Address ADV OR 1.1
1.9	21 Oct 2015	TG	Correct Audit events
2.0	21 Oct 2015	TG	Corrected build number and added Windows 2008
2.1	22 Oct 2015	TG	Corrected Audit Event Codes

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Overview

1 The CloudMask Engine (CloudMask) is a Cloud Computing security product that enables users to protect their confidential data while leveraging public and private cloud applications.

2 This Security Target (ST) defines the CloudMask Engine v2.0 Target of Evaluation (TOE) for the purposes of Common Criteria (CC) evaluation.

3 Whilst CloudMask offers a wide range of features, the TOE is constrained to the security features outlined in section 2.3.

1.2 Identification

Table 1: Evaluation identifiers

Target of Evaluation	CloudMask Engine v2.0 Build 610
Security Target	CloudMask Engine v2.0 Security Target, v2.1

1.3 Conformance Claims

4 This ST supports the following conformance claims:

a) CC version 3.1 Release 4

b) CC Part 2 conformant

c) CC Part 3 conformant

d) Evaluation Assurance Level 2 (EAL2) conformant

1.4 Terminology

Table 2: Terminology

Term	Definition
Application	Cloud application under control of the TOE.
Application Administrator	Administrator of cloud applications in the Cloud
Application Sharing Rules	Rules governing how user data in cloud applications is shared between users (i.e. when greater than one user has access to the same application and set of data).
3 DES	Triple DES (Data Encryption Standard)
AES	Advanced Encryption Standard
CA	Certificate Authority

CC	Common Criteria
Security Officer	Administrator of the TOE
Cloud Service Provider (CSP)	A third party provider of cloud computing services, including software as a service (SaaS) as defined by NIST in SP 800-145.
Cryptographic Keys	Keys generated by the cryptographic engine, used for cryptographic processes described in ST section 6.
EAL	Evaluation Assurance Level
Element Rules	Rules that define the web data elements to be encrypted and the applicable algorithm used to encrypt/decrypt data.
Encrypted Data	User data encrypted by the TOE.
Group	More than one user granted access to the same cloud application user data.
Information Assets	User data under protection of the TOE.
JSON	File type that uses human-readable text to transmit data objects consisting of attribute-value pairs. It is used primarily to transmit data between a server and web application, as an alternative to XML.
Mask	Encoded tokenized data, as to adhere to syntax requirements of the cloud application.
MSCAPI	Microsoft Cryptographic Application Programming Interface
PKI	Public Key Infrastructure
PP	Protection Profile
Recipient list	A list maintained by the CloudMask Manager for user(s) to access user data.
RSA	Rivest, Shamir, Adleman public key crypto system
TOE	Target of Evaluation
Token	Data generated to replace user data in cloud applications.
User	User of the TOE
User Data	Data belonging to an end user of the TOE, associated with one or more cloud applications.

2 TOE Description

2.1 Type

5 The TOE is a Cloud Computing security product installed as a software application on end-user devices to ensure confidentiality of user data in cloud applications.

2.2 Usage

6 The TOE performs data detection and protection functions that enable users to protect their confidential data while leveraging public and/or private cloud applications. The TOE is comprised of software that integrates with a web browser. Depending on the platform the TOE is implemented on, it can be a software application (for desktops, and other personal computers) or a mobile app (for mobile devices).
The evaluated configuration is limited to software application.

7 The CloudMask Engine is pre-configured with a variable for the distinguished name of a Security Officer role, which allows for the installation and configuration of the Engine. The assigned Security Officer is responsible for installation, configuration, and administration of the TOE including user enrolment.

8 The TOE requires a cryptographic engine to be implemented in the environment to perform cryptographic functions as described below. The evaluated configuration uses Entrust ESP v9.2 PKI as it's cryptographic engine (Crypto Engine)*. Users are enrolled through Entrust ESP to create a digital ID, at the discretion of the designated Security Officer, and are authenticated to the TOE using the Entrust ESP/MSCAPI interface. Upon successful registration, end users are then granted access by the Security Officer to Cloud applications under the protection of the TOE.

9 TOE users then transparently use the TOE to protect cloud application data in accordance with the descriptions below.

10 CloudMask Manager is also a mandatory component, either implemented in CloudMask Cloud services, or as a Cloud installation internal to the organization. It provides limited management functions as described below in 2.4.2 b).

** CloudMask can be implemented without a PKI in the environment, in which case it relies on self-signed certificates generated by the Crypto Engine. This is not part of the evaluated configuration.*

2.2.1 CloudMask Engine (TOE)

11 **Tokenization**

12 CloudMask Engine (CloudMask) works transparently by intercepting application data before it is transmitted to the cloud and replacing it with a random token in a process called tokenization. The tokenized data is meaningless unless viewed by an authorized CloudMask user. To achieve this, CloudMask intercepts application data and generates tokens representing the data and in parallel encrypts the original data (ciphered data). Encryption is performed by Entrust ESP v9.2, where instructions on algorithms and key lengths are provided by the TOE to the crypto engine by way of code injection.

13 The tokenized data is then encoded to conform with the expected syntax of the respective cloud application, producing a "mask". The mask is transmitted to the cloud application and the mask and associated ciphered data is sent to a CloudMask server (public or private, known as CloudMask Manager) for storage. Encryption keys used on Cloud Application Data are also distributed to the Manager for storage, where the symmetric keys used to encrypt the data are encrypted to each recipient's public key. CloudMask Manager is a mandatory requirement for the TOE environment and is further defined in section 2.4.2.

14 When a user accesses their data in the future, the same process described above is reversed - CloudMask intercepts the request and retrieves the Ciphered mask associated with the mask from the CloudMask server.

15 The method used to generate tokens and consequently the mask ensures that:

- There is no mathematical relationship between original user data and its token (tokens are sequence numbers).
- Tokens are randomly selected while conforming to web application syntax requirements.
- Same user data yields randomly different tokens over time.

Configuration

The CloudMask Engine defines policies and configuration data. Configuration data is created at the Engine by the designated Security Officer, via a HTML5 GUI injected directly by the Engine, where it is signed by the Security Officer and pushed to entities in the environment.

Policies are generated at the Engine using JSON records and signed by the Security Officer, and include data handing rules, user identity attributes, groups, and access control lists.

16 **System Requirements**

17 CloudMask Engine runs as a software application available on the following list of Operating System/Web Browser combinations:

- Microsoft Windows 7 with Microsoft Internet Explorer 11
- Microsoft Windows 2008 R2 with Microsoft Internet Explorer 11

18 **Deployment**

19 CloudMask can be deployed in a scalable number to end user devices, allowing one to many installations as required by an organization. CloudMask can be installed and configured by the Security Officer via the CloudMask Manager.

20 CloudMask Engine for Internet Explorer 11 is a Windows executable file, installed following Microsoft installation guidelines. The executable, Windows installer (.msi), and other components are digitally signed by CloudMask.

21 The distribution of the executable leverages existing enterprise package management tools or it can be downloaded from the web.

22 The CloudMask Manager is a collective name for various components outside of the TOE that are responsible for data storage including customizations for data handling. The CloudMask Manager does not provide security-enforcing functionality, however the Engine must be deployed with the Manager component. CloudMask Manager

can be deployed in either a private cloud or on CloudMask cloud servers. Instructions on deployment are described in Section 2.4.1 referenced document – Deployment Guide v1.1.

23 Figure 1 below depicts the logical TOE scope and interactions with the cryptographic engine and CloudMask Manager in the environment.

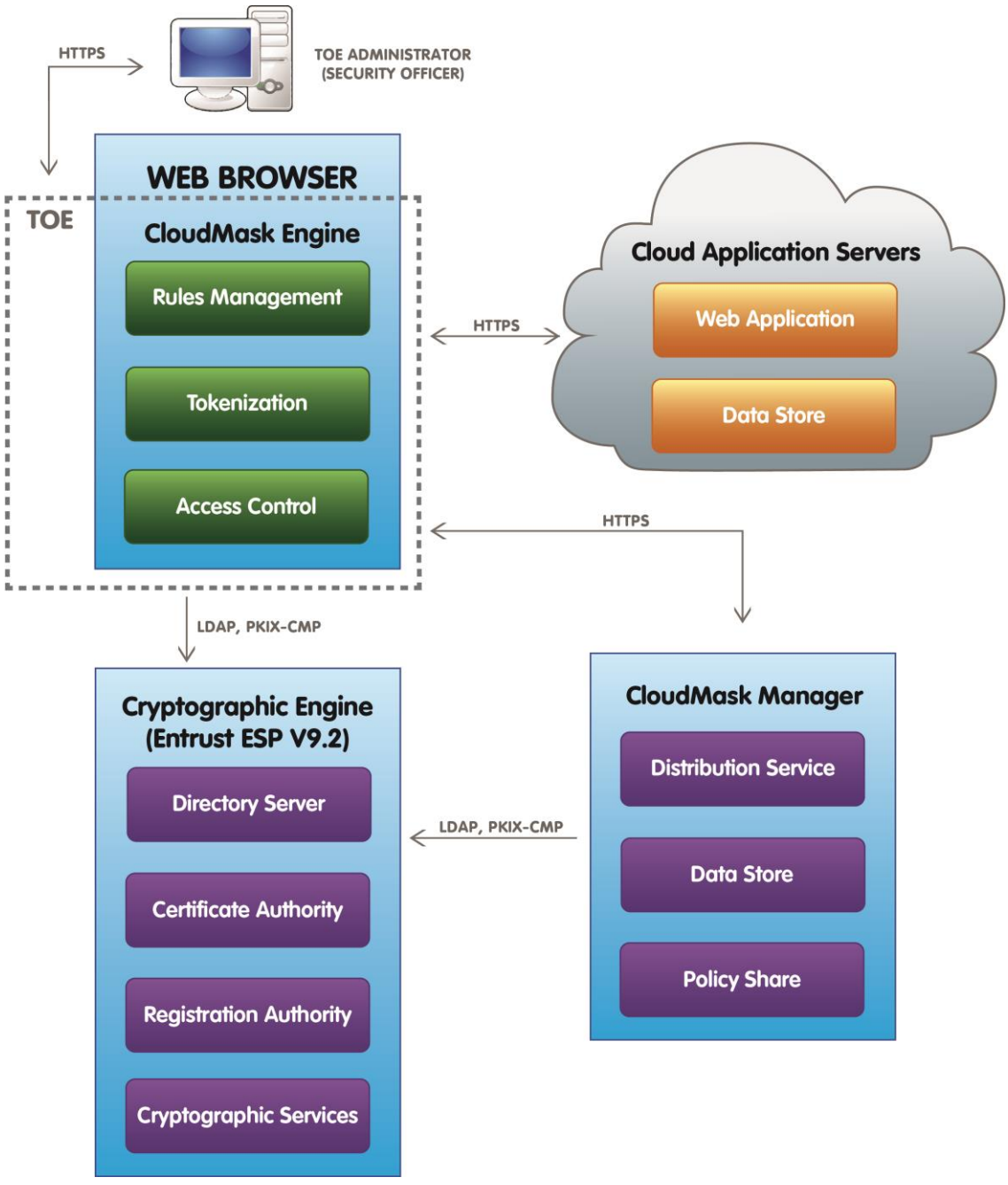

Figure 1: TOE Architecture

2.3 Security Functions

24 The TOE provides the following security functions:

a) **Data Handling Rules.** Creation and enforcement of application data handling rules in JSON records which instruct CloudMask Engine on what and how to encrypt/decrypt various application fields (default protects all textual fields).

b) **Data Tokenization.** Tokenization of encrypted data (generating the Mask – encoded tokens)

c) **Access Control Policies.** Creation and enforcement of access control policies in JSON records, enforced by using the correct user and group user public keys

d) **Protected Communications.** Invocation of secure communication channels (HTTPS)

e) **Information Flow Control.** Enforcement of information flow: Mask -> Cloud Application, Mask -> CloudMask server and vice versa

f) **Trusted update.** Application code is digitally signed for validation by the install environment.

2.4 Physical Scope

25 The TOE is comprised of the following software:

a) CloudMask Engine (A software application)

2.4.1 Guidance Documents

26 The TOE includes the following guidance documents:

a) CloudMask Enterprise Knowledge Base for version 2.0 (https://support.cloudmask.com)

2.4.2 Non-TOE Components

27 The TOE operates with the following components in the environment:

a) **Cryptographic Engine (Crypto Engine).** The TOE uses MS CryptoAPI to call on cryptographic services in the environment through a cryptographic service provider (cryptographic engine). The cryptographic engine (Crypto Engine) is called by the TOE to generate cryptographic keys, issue and maintain certificates, manage keys, and perform cryptographic operations. Consumers must implement a FIPS 140-2 validated crypto engine, and the evaluated configuration uses Entrust ESP v9.2.

b) **CloudMask Manager.** The CloudMask Manager is a collective name for various components outside of the TOE that are responsible for data storage including customizations for data handling. Management functions are as follows:

- **Distribution Service** Acts as a repository for users to install and update the CloudMask Engine using a simple web page. This is an optional component since the distribution of CloudMask Engine may be performed

using in-place software distribution tools. In all cases, the integrity of the software is ensured through software code signing.

- **Data Store** Provides CloudMask engine with the services necessary to store tokens, encrypted data, and encrypted symmetric keys after the TOE performs data tokenization. Tokens contain record ID (sequence number) that is used by the Engine to fetch encrypted data and encrypted keys. Storage is external to the TOE.
- **Policy Store**: Provides CloudMask engine with the necessary services to store signed access control policies, users, groups, and data handling rules.

2.5 Logical Scope

28 The logical scope of the TOE comprises the security functions defined in section 2.3.

3 Security Problem Definition

3.1 Threats

29 Table 3 identifies the threats addressed by the TOE.

Table 3: Threats

Identifier	Description
T.APP_ADMIN_EXPOSE	User data is disclosed to application administrator at the Cloud Service Provider (CSP), through malicious insider, external attacker, or legal disclosure through third party.
T.KEY_REVEAL	Information assets or cryptographic keys for those assets are revealed to the cloud mask administrators, leading to unauthorized disclosure.
T.TOKEN_EXPLOIT	Tokens masking user data can be exploited by a malicious third party by way of reversible relation to user data, leading to unauthorized disclosure.
T.RESIDUAL_DATA	When a contract ends with a CSP or an entity ceases operation, data may still remain at the CSP and not be securely destroyed, leading to unauthorized disclosure.
T.FOREIGN	Information assets residing with a CSP in a foreign jurisdiction may not be subject to controls or Organizational Security Policies (OSPs) required in country of origin, leading to unauthorized disclosure of user data to foreign entities.
T.PHYSICAL	Physical security on CSP premises could be compromised by malicious third party, leading to unauthorized access to user data.
T.INTERCEPT	User data is intercepted by and disclosed to a malicious third party between the TOE and the CSP.
T.AUDIT_DATA	CloudMask administrator changes encrypted data undetected.
T.UPDATE_EXPLOIT	Updates to the TOE are compromised by a malicious third party to reveal user data.

3.2 Organizational Security Policies

30 Table 4 identifies the Organizational Security Policies (OSPs) that are addressed by the TOE.

Table 4: OSPs

Identifier	Description
OSP.CRYPTO_ENGINE	A cryptographic engine must be implemented to interface with the TOE to generate, distribute, and maintain cryptographic keys and digital certificates, perform cryptographic operations, and provide mechanisms for identifying and authenticating end users, with such requirements being evaluated in scope of a FIPS 140-2 validated engine.
OSP.DATA_STORE	A database will be provisioned for the purposes of storing all cipher user data, encrypted symmetric keys, tokens, signed configuration rules, and signed access control policies.

3.3 Assumptions

31 Table 5 identifies the assumptions related to the TOE's environment.

Table 5: Assumptions

Identifier	Description
A.TRUSTED_ADMIN	TOE Administrators (Security Officer) are trusted and available to perform management functions.

4 Security Objectives

4.1 Objectives for the Operational Environment

32 Table 6 identifies the objectives for the operational environment.

Table 6: Operational environment objectives

Identifier	Description
OE.CRYPTO_ENGINE	Cryptographic keys digital certificates/signatures will be generated and managed through the Crypto Engine (either physically on the device or through external CA), and cryptographic operations performed, logically external to the TOE.
OE.ACCOUNT_LOCKING	Crypto Engine Provider will provide session security account locking after Security Officer-specified number of unsuccessful login attempts.
OE.ID_AUTH	Crypto Engine Provider will provide user identification and authentication for the TOE.
OE.MANAGEMENT	A trusted CloudMask Security Officer will be made available by the user organization to provide administrative services to manage rules and access control directly to CloudMask Engine.
OE.TIMESTAMPS	Windows 7 and Windows 2008 R2 real-time timestamps will provide timestamping.
OE.DATA_STORE	The CloudMask Manager will provide a data store for storing all cipher user data, encrypted symmetric keys, tokens, signed rules, and signed access control policies.

4.2 Objectives for the TOE

33 Table 7 identifies the security objectives for the TOE.

Table 7: Security objectives

Identifier	Description
O.PREVENT_ADMIN_ACCESS	The TOE shall prevent CloudMask Administrators from accessing cleartext user data.
O.PREVENT_CLOUD_ACCESS	The TOE shall prevent entities in the Cloud from accessing cleartext user data.

Identifier	Description
O.ACCESS_CONTROL	The TOE shall enforce user and group access controls to applications cleartext data.
O.VERIFIABLE_UPDATE	The TOE shall be able to invoke the Windows Installer services to verify updates.
O.TOKENIZE	The TOE shall replace user data with tokens and enforce rules to ensure user application data is tokenized and encrypted before it leaves the TOE.
O.DATA_INTEGRITY	The TOE shall associate data with digital certificates to provide data integrity.
O.SECURE_ADMIN	The TOE shall authenticate CloudMask Engine Security Officers and record a log of their actions.
O.SECURE_COMMS	The TOE will enforce secure (HTTPS) communications between itself, and CloudMask Manager.

5 Security Requirements

5.1 Conventions

34 This document uses the following font conventions to identify the operations defined by the CC:

a) **Assignment.** Indicated with italicized text.

b) **Refinement.** Indicated with bold text and strikethroughs.

c) **Selection.** Indicated with underlined text.

d) **Assignment within a Selection:** Indicated with italicized and underlined text.

e) **Iteration.** Indicated by appending the iteration number in parenthesis, e.g., (1), (2), (3).

35 Explicitly stated SFRs are identified by having a label 'EXT' after the requirement name for TOE SFRs.

5.2 Extended Components Definition

36 The ST does not incorporate any extended components.

5.3 Functional Requirements

Table 8: Summary of SFRs

Family	Requirement	Title
Security Audit	FAU_GEN.1	Audit data generation
	FAU_GEN.2	User identity association
Identification & Authentication	FIA_UAU.2	User authentication before any action
	FIA_UID.2	User identification before any action
User Data Protection	FDP_ACC.2	Complete access control
	FDP_ACF.1	Security attribute based access control
	FDP_ETC.2	Export of user data with security attributes
	FDP_IFC.1	Subset information flow control
	FDP_IFF.1	Simple security attributes
	FDP_ITC.2	Import of user data with security attributes
	FDP_UCT.1	Basic data exchange confidentiality

Family	Requirement	Title
Security Management	FMT_MOF.1	Management of security functions behaviour
	FMT_MSA.1	Management of security attributes
	FMT_MSA.2	Secure security attributes
	FMT_MSA.3	Static attribute initialisation
	FMT_SMF.1	Specification of management functions
	FMT_SMR.1	Security roles
Protection of the TSF	FPT_ITC.1	Inter-TSF confidentiality during transmission
	FPT_TDC.1	Inter-TSF basic TSF data consistency
Trusted Path/Channels	FTP_ITC.1	Inter-TSF trusted channel

5.3.1 Security Audit (FAU)

FAU_GEN.1 Audit Data Generation

Hierarchical to: No other components.

Dependencies: FPT_STM.1 Reliable time stamps*

** FPT_STM.1 dependency is not fulfilled. The TOE timestamps audit records, and relies on the Windows 7 and Windows 2008 R2 real-time timestamps for timestamping audit data. See OE.TIMESTAMPS.*

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

a) Start-up and shutdown of the audit functions;

b) All auditable events for the not specified level of audit; and

c) *All administrative actions*;

d) *Auditable events listed in the table below*].

Code	Description	Additional Properties
APP_CREATED	A new Application has been created	
APP_UPDATED	An Application Rule has been changed	• Internal Application ID

Code	Description	Additional Properties
APP_VERIFIED	Application Rule signature has been verified	• Internal Application ID
APP_INVALID	Application Rule signature failed validation	• Internal Application ID
GROUP_CREATED	A new Group has been created	
GROUP_UPDATED	A Group has been changed	• Internal Group ID
GROUP_VERIFIED	Group signature has been verified	• Internal Group ID
GROUP_INVALID	Group signature failed validation	• Internal Group ID
USER_ERROR	User login or enrolment Failure	• Reason for failure • DN of the certificate
USER	User login	• DN of the certificate
USER_INVALID	Invalid user credentials detected during attempt to encrypt data	• Email of invalid user • Internal ID of user • DN of the invalid certificate • Reason for not trusting the certificate
DATA_CREATED	Encrypted record for one or more user	• Unique record ID • A list of user emails the data is encrypted for
DATA_UPDATED	Record re-encrypted with modified data	• Unique record ID
DATA_SHARED	Data encryption key shared with one or more user	• A list of user emails the data is encrypted for
DATA_INVALID	No signature, or an invalid signature was detected on the record	• Reason for the signature error

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [*additional details specified in the above table*].

FAU_GEN.2 User Identity Association

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation
FIA_UID.1 Timing of identification

FAU_GEN.2.1 For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

5.3.2 Identification and Authentication (FIA)

FIA_UAU.2 User authentication before any action

Hierarchical to:	FIA_UAU.1 Timing of authentication
Dependencies:	FIA_UID.1 Timing of identification
FIA_UAU.2.1	The **TSF** shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

FIA_UID.2 User identification before any action

Hierarchical to:	FIA_UID.1 Timing of identification
Dependencies:	No dependencies.
FIA_UID.2.1	The **TSF** shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

5.3.3 User Data Protection (FDP)

FDP_ACC.2 Complete access control

Hierarchical to:	FDP_ACC.1 Subset access control
Dependencies:	FDP_ACF.1 Security attribute based access control
FDP_ACC.2.1	The TSF shall enforce the *CloudMask access control SFP* on *user(s) and user(s) application data* and all operations among subjects and objects covered by the SFP.
FDP_ACC.2.2	The TSF shall ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.

FDP_ACF.1 Security attribute based access control

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1 The TSF shall enforce the *CloudMask access control SFP* to objects based on the following:

- *Subjects: Application User(s)*
- *Objects: Application User(s) Data*
- *Object Attributes: Digital Certificate, Storage encryption keys*

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: *The user has the signed digital certificate for the application data which authenticates the user. User data encryption keys can then be used by the TOE and the Security Officer to read and/or write user data. If authentication is unsuccessful, access to encrypted data is denied.*

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *no additional rules*.

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *no additional rules.*

FDP_ETC.2 Export of user data with security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or

FDP_IFC.1 Subset information flow control]

FDP_ETC.2.1 The TSF shall enforce the *CloudMask access control SFP(s) and the information flow control SFP(s)* when exporting user data, controlled under the SFP(s), outside of the TOE.

FDP_ETC.2.2 The TSF shall export the user data with the user data's associated security attributes.

FDP_ETC.2.3 The TSF shall ensure that the security attributes, when exported outside the TOE, are unambiguously associated with the exported user data.

FDP_ETC.2.4 The TSF shall enforce the following rules when user data is exported from the TOE: *no additional exportation control rules*.

FDP_IFC.1 Subset information flow control

Hierarchical to: No other components.

Dependencies: FDP_IFF.1 Simple security attributes

FDP_IFC.1.1 The TSF shall enforce the *CloudMask information flow control SFP* on:

- *Subjects: User Web Browser, Cloud Application*
- *Information: Cloud Application Data*
- *Operations: Read, Write*

FDP_IFF.1 Simple security attributes

Hierarchical to: No other components.

Dependencies: FDP_IFC.1 Subset information flow control

FMT_MSA.3 Static attribute initialisation

FDP_IFF.1.1 The TSF shall enforce the *CloudMask application data access information flow control SFP* based on the following types of subject and information security attributes: *subjects and information controlled under the indicated SFP, and for each, the security attributes below:*

Subjects: Browser, Cloud Application

Information: URL, Application data

Security Attributes: Element rules, Recipient List, Digital Certificate.

FDP_IFF.1.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

- *If the user requesting data through the cloud application has a digital certificate that matches the recipient list for the data, the TSF applies element rules to perform mappings to get the data, and vice versa for writing data to the cloud application.*

FDP_IFF.1.3 The TSF shall enforce no *additional information flow control SFP rules*.

FDP_IFF.1.4	The TSF shall explicitly authorise an information flow based on the following rules: *no additional rules*.
FDP_IFF.1.5	The TSF shall explicitly deny an information flow based on the following rules: *no additional rules.*

FDP_ITC.2 Import of user data with security attributes

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path] FPT_TDC.1 Inter-TSF basic TSF data consistency
FDP_ITC.2.1	The TSF shall enforce the *CloudMask access control SFP and information flow control SFP(s)* when importing user data, controlled under the SFP, from outside of the TOE.
FDP_ITC.2.2	The TSF shall use the security attributes associated with the imported user data.
FDP_ITC.2.3	The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.
FDP_ITC.2.4	The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.
FDP_ITC.2.5	The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: no *additional importation control rules*.

FDP_UCT.1 Basic data exchange confidentiality

Hierarchical to:	No other components.
Dependencies:	[FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path] [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]
FDP_UCT.1.1	The TSF shall enforce the CloudMask *access control SFP and/or CloudMask information flow control SFP* to *transmit, receive* user data in a manner protected from unauthorised disclosure.

5.3.4 Security Management (FMT)

FMT_MOF.1 Management of security functions behaviour

Hierarchical to:	No other components.
Dependencies:	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
FMT_MOF.1.1	The TSF shall restrict the ability to determine the behaviour of, disable, enable, modify the behaviour of the functions *outlined in FMT_SMF.1* to the *Security Officer*.

FMT_MSA.1 Management of security attributes

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
FMT_MSA.1.1	The TSF shall enforce the *CloudMask access control SFP(s), CloudMask information flow control SFP(s)* to restrict the ability to *change_default, query, modify, delete, and no other operations,* the security attributes *listed in FDP_IFF.1* to *the Security Officer*.

FMT_MSA.2 Secure security attributes

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles
FMT_MSA.2.1	The TSF shall ensure that only secure values are accepted for *digital certificates, Element Rules, encryption algorithms*.

FMT_MSA.3 Static attribute initialisation

Hierarchical to:	No other components.
Dependencies:	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles
FMT_MSA.3.1	The TSF shall enforce the *CloudMask access control SFP, CloudMask information flow control SFP* to provide restrictive *and no other property* default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2	The TSF shall allow the *Security Officer* to specify alternative initial values to override the default values when an object or information is created.

FMT_SMF.1 Specification of Management Functions

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FMT_SMF.1.1	The TSF shall be capable of performing the following management functions: • *Manage group membership and group key lifetime/rules* • *Define which applications to protect* • *Manage application element rules, which defines the web data elements to be encrypted and the applicable algorithm.* • *Manage access control through application sharing rules.* • *Initiate re-encryption of data (officers are always a recipient on every database record and must re-encrypt data in the event of viewing).*

- *Initiate encryption of symmetric keys for given user certificate.*

FMT_SMR.1 Security roles

Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification
FMT_SMR.1.1	The TSF shall maintain the roles *Security Officer, Authorized User*.
FMT_SMR.1.2	The TSF shall be able to associate users with roles.

5.3.5 Protection of the TSF (FPT)

FPT_ITC.1 Inter-TSF confidentiality during transmission

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_ITC.1.1	The TSF shall protect all TSF data transmitted from the TSF to another trusted IT product from unauthorised disclosure during transmission.

FPT_TDC.1 Inter-TSF basic TSF data consistency

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_TDC.1.1	The TSF shall provide the capability to consistently interpret *masked user cloud application data* when shared between the TSF and another trusted IT product.
FPT_TDC.1.2	The TSF shall use *Element Rules and Mask lookup based on field ID,* when interpreting the TSF data from another trusted IT product.

5.3.6 Trusted Path/Channels (FTP)

FTP_ITC.1 Inter-TSF trusted channel

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FTP_ITC.1.1	The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.
FTP_ITC.1.2	The TSF shall permit *the TSF, CloudMask Manager component* to initiate communication via the trusted channel.
FTP_ITC.1.3	The TSF shall initiate communication via the trusted channel for *users reading or writing cloud application data under control of the TSF*.

5.4 Assurance Requirements

37 The TOE security assurance requirements, summarized in Table 9, are commensurate with EAL2.

Table 9: Assurance Requirements

Assurance Class	Components	Description
Development	ADV_ARC.1	Security Architecture Description
	ADV_FSP.2	Security-enforcing Functional Specification
	ADV_TDS.1	Basic Design
Guidance Documents	AGD_OPE.1	Operational User Guidance
	AGD_PRE.1	Preparative User Guidance
Life Cycle Support	ALC_CMC.2	Use of a CM System
	ALC_CMS.2	Parts of the TOE CM Coverage
	ALC_DEL.1	Delivery Procedures
Security Target Evaluation	ASE_CCL.1	Conformance Claims
	ASE_ECD.1	Extended Components Definition
	ASE_INT.1	ST Introduction
	ASE_OBJ.2	Security Objectives
	ASE_REQ.2	Derived Security Requirements
	ASE_SPD.1	Security Problem Definition
	ASE_TSS.1	TOE Summary Specification
Tests	ATE_COV.1	Evidence of Coverage
	ATE_FUN.1	Functional testing
	ATE_IND.2	Independent Testing - sample
Vulnerability Assessment	AVA_VAN.2	Vulnerability Analysis

6 TOE Summary Specification

6.1 Data Handling Rules

Related SFRs:	FDP_ACC.2, FDP_ACF.1, FDP_ETC.2, FDP_IFF.1, FDP_ITC.2, FDP_UCT.1, FMT_MOF.1, FMT_MSA.1, FMT_MSA.3, FMT_SMF.1, FMT_SMR.1, FPT_TDC.1,

38 CloudMask works with any number of cloud applications by creating rules to ensure data is handled in accordance with cloud application protocol for data handling.

39 When a user creates, modifies, or requests cloud application data, it's the application data handling rules that instruct CloudMask Engine on what and how to encrypt/decrypt various application fields, which by default encrypts, tokenizes and masks all textual fields. Custom rules may be defined by the CloudMask Security Officers in order to override default behaviour and values. At a low level, CloudMask intercepts the object data by looking up Element Rules. An Element Rule contains conditions on what web element to intercept and what rules to apply against it based on Usage, Tokenization and EventType fields. The Element Rules are part of an Application policy object digitally signed by a Security officer.

40 The management functions for data handling rules are accessed through a Web GUI by the Security Officer, using code injected by the Engine, pulling information from JSON records. Policy objects (User, Group and Application) are digitally signed by the CloudMask Security Officer using the CloudMask Engine. The signed objects are first verified by the Engine before applying any data handling rules.

6.2 Data Tokenization

Related SFRs:	FDP_ETC.2, FDP_UCT.1, FPT_TDC.1, FMT_MSA.2

41 CloudMask Engine provides a mechanism to ensure sensitive data cannot be exploited, either in transit or in storage. That process, which is described here as tokenization, takes application data from the client side of a cloud application and performs a series of steps to ultimately replace data with tokens.

42 Client side cloud application data is first encrypted by the engine and encrypted data passed to the TSF as described below:

43 Encryption of user data is invoked by the TOE and parameters specified at the TOE for encryption options. Encryption/decryption is performed at a crypto engine in the TOE environment, specifically Entrust Entelligence Security Provider v9.2. The encryption of client side cloud application data is implemented using a mix of symmetric and public/private key pairs by the Crypto Engine, with instructions by the TOE as follows:

- User data is encrypted using a random symmetric key, with two options for key expiration as configured at the TOE.

i. A random symmetric key is by default generated for every new record, with the same key used for future edits.

ii. If configured other than default, one key can be used for the lifetime of data (group key). The expiration period of keys can be user defined for any specified period of time.

- The symmetric key is encrypted using the user's public key, as well as other recipients' public keys.
- If the TOE is configured for group key, the TOE instructs the crypto engine to have the symmetric key encrypted using each group member's public key. This is only done when the group symmetric key is about to expire.
- Encryption algorithms are configurable by the TOE and are implemented directly by the Crypto Engine. The TOE is configured to invoke AES-256 encryption by default, but can be changed by the Security Officer to use 3DES.
- The symmetric key is never stored in clear text on any location and is only kept momentarily in the user's memory to perform necessary operations. They are exported from the crypto engine after being encrypted using the recipients' public keys and stored on the Manager.
- CloudMask releases the handles to the keys within the crypto engine, and once the release occurs the crypto engine is responsible for securely deleting any memory blocks.

44 The encrypted data is then replaced by a token, in a process called tokenization. The token is composed of sequence numbers uniquely identifying the record and field, so that these can be referenced and retrieved when the client application makes a request. Each record has a unique sequence number, assigned based on the order of creation, and accordingly not related to the content of the record. The field ID is a number generated to identify the exact field and its version within the record. It is also assigned as needed, when the user is creating or editing fields.

45 A mask is then created by encoding the token, and optionally some of the encrypted data. The encoding converts the token into a format that respects the application syntax rules of the cloud application. The encoding itself does not add or remove security of the data and is purely a formatting issue. It is intended to preserve the format syntax restriction that may be required by a given application.

46 When a user requests information from a cloud application, the process is reversed, where record and field IDs are matched up on request from the User, and used to retrieve encrypted data and associated encrypted keys from the Data Store, to ultimately have the data decrypted.

6.3 Access Control Policies

Related SFRs:	FAU_GEN.1, FAU_GEN.2, FIA_UAU.2, FIA_UID.2, FDP_ACC.2, FDP_ACF.1, FDP_ETC.2, FDP_IFF.1, FDP_ITC.2

47 CloudMask Engine safeguards user data by restricting access to that data based on user-configured access control policies. Data access is either restricted to an individual user or to a group of users. The enforcement of access control policies is

through the use of user and group user public keys to encrypt the symmetric encryption key.

48 A designated CloudMask Security Officer performs initial setup and ongoing maintenance to the CloudMask access control policies. The Security Officer will install the TOE, and then add users to the system. The users must be setup with trusted certificates, and if no certificate exists, cannot login to the TOE. Trust is verified through the crypto engine verification of the trust chain of the certificate. The Security Officer may also sign the User object to explicitly trust the given certificate. Any required user attributes that are not available in the certificate (e.g. email, phone number), must also be signed by the Security Officer as part of the user object.

49 To ensure a regular user cannot be elevated to the Security Officer role without authorization, the Engine independently verifies the digital signature against the officer's trusted certificate. The officer's certificate is configured during the TOE installation. The certificate may be packaged with the install or using its distinguished name, may be used to fetch and validate the certificate using the crypto engine.

50

51 **Application Access**

52 Access restrictions are part of Application policy objects defined by the security officer through the Web GUI, injected by the TOE. The Application policies, including Access Rules and Element Rules, can be configured by the Security Officer through the Web GUI. Access rules define the Cloud application records that a given user has access to under control of the TOE, and the element rules define how various application fields should be treated with respect to tokenization.

53

54 **Identification and Authentication**

55 It is the crypto engine (not part of TOE) that manages the identification and authentication process, and provides feedback to the end user. CloudMask Engine's role in the process of access control is to request these services from the environment and enforce access control policies to data upon identification and authentication. When a user makes a request to gain access to applications and data, CloudMask Engine calls the Crypto engine, determining if the user is logged in, locked out or timed out.

56 The CloudMask Manager stores a list of recipients (identified by certificate and/or certificate distinguished name), which is maintained to control access to objects under control of the TSF. In the event of a compromise, the data may only be decrypted by the authorized recipient, given that recipient's symmetric key is encrypted for specific private keys (that of data being requested). The TOE makes calls on behalf of the user to the Manager recipient list of a given record to verify those user credentials match the recipient list before returning any data.

57 Group access control is maintained by Group and Sharing Rules. The group describes a list of members and is digitally signed by a trusted security officer. The Sharing Rule specifies the conditions when data should be shared with a given group. The Sharing Rule is part of the Application configuration object, which is also digitally signed.

58 Access to the TSF is logged in accordance with FAU_GEN.1.

6.4 Protected Communications

Related SFRs:	FDP_UCT.1, FPT_ITC.1, FTP_ITC.1

59 All TSF data communicated between the TOE and the CloudMask Manager uses an HTTPS secured connection. The configuration can be managed locally by the user through the client web browser, where a secure channel (HTTPS) is invoked, and is enforced through the TOE for communication between these various entities. Communications with cloud applications rely on the Web browser to enforce HTTPS.

60 Only TLS v1.0 or higher is permitted to be invoked in the evaluated configuration.

61 The engine calls the browser to invoke an HTTPS session to the configured manager URL. The server certificate is validated as trusted using standard SSL certificate trust chain provided by the browser. The client identity is verified by the server using a custom authentication header, where by the TOE calls the crypto engine on behalf of the user to sign a server nonce and a client nonce.

6.5 Information Flow Control

Related SFRs:	FDP_ACC.2, FDP_ACF.1, FDP_ETC.2, FDP_IFC.1, FDP_IFF.1, FDP_ITC.2, FIA_UAU.2, FIA_UID.2, FDP_UCT.1, FMT_MSA.3, FMT_SMF.1, FTP_ITC.1

62 Information flow is controlled by the TOE at the application layer for transmission of data from the TOE as a Mask to Cloud Application, and of encrypted data, encrypted keys, tokens, and signed policies/rules to the CloudMask Manager for data storage, and vice versa.

63 **Application Layer**

64 Cloud application data at the Engine (TOE) is encrypted on the TOE, tokenized then encoded to produce a mask. CloudMask Engine intercepts application data and generates tokens representing the data (Mask) and then invokes the crypto engine to encrypt the original data (Ciphered Mask), preventing unencrypted data from leaving the TOE. The Mask is transmitted to the cloud application and the Mask and associated Ciphered Mask are sent to a CloudMask server (public or private, known as CloudMask Manager) for storage.

65 When the user accesses their data in the future, the same process described above is reversed - CloudMask Engine intercepts the request and retrieves the Ciphered Mask associated with the Mask from the CloudMask server.

66 If the user requesting data through the cloud application has a digital certificate that matches the recipient list for the data, the TSF applies element rules to perform mappings to get the data, and vice versa for writing data to the cloud application. Only when the user has the credentials required will the TOE permit the decryption of data retrieved from the server.

6.6 Trusted Update

Related SFRs:	FAU_GEN.1, FAU_GEN.2, FIA_UAU.2, FIA_UID.2, FMT_MOF.1, FMT_MSA.1, FMT_MSA.2, FMT_SMF.1, FTP_ITC.1

67 Software update files are digitally signed with CloudMask code signing key - X.509 / RSA 2048 bit key by the vendor prior to distribution. The TOE verifies digital signatures prior to installing updates and aborts if signature verification fails. The verification is performed by using the Operating System / Browser Environment installation utilities (e.g. standard Windows Installer).

7 Rationale

7.1 Conformance Claim Rationale

68 There TOE does not claim conformance to a Protection Profile.

7.2 Security Assurance Requirements Rationale

69 EAL2 was chosen to provide a low to moderate level of assurance that is consistent with good commercial practices. As such, minimal additional tasks are placed upon the vendor assuming the vendor follows reasonable software engineering practices and can provide support to the evaluation for design and testing efforts. The chosen assurance level is appropriate with the threats defined for the environment. At EAL2, the TOE will have undergone a search for obvious flaws to support its introduction into the non-hostile environment.

7.3 Security Objectives Rationale

Table 10: Security Objectives Mapping

	T.APP_ADMIN_EXPOSE	T.KEY_REVEAL	T.TOKEN_EXPLOIT	T.RESIDUAL_DATA	T.FOREIGN	T.PHYSICAL	T.INTERCEPT	T.AUDIT_DATA	T.UPDATE_EXPLOIT	OSP.CRYPTO_ENGINE	OSP.DATA_STORE	A.TRUSTED_ADMIN
O.PREVENT_ADMIN_ACCESS		X						X				
O.PREVENT_CLOUD_ACCESS	X		X	X	X	X						
O.ACCESS_CONTROL	X	X										
O.VERIFIABLE_UPDATE									X			
O.TOKENIZE	X		X	X	X	X	X					
O.DATA_INTEGRITY								X				
O.SECURE_COMMS							X		X			
O.SECURE_ADMIN								X				
OE.CRYPTO_ENGINE										X		

	T.APP_ADMIN_EXPOSE	T.KEY_REVEAL	T.TOKEN_EXPLOIT	T.RESIDUAL_DATA	T.FOREIGN	T.PHYSICAL	T.INTERCEPT	T.AUDIT_DATA	T.UPDATE_EXPLOIT	OSP.CRYPTO_ENGINE	OSP.DATA_STORE	A.TRUSTED_ADMIN
OE.ACCOUNT_LOCKING										X		
OE.ID_AUTH										X		
OE.MANAGEMENT												X
OE.TIMESTAMPS								X				
OE.DATA_STORE											X	

Table 11: Suitability of Security Objectives

Element	Justification
T.APP_ADMIN_EXPOSE	**O.PREVENT_CLOUD_ACCESS.** Prevents entities in the cloud accessing cloud application data. **O.ACCESS_CONTROL.** Access controls limit parties that can access application data. **O.TOKENIZE.** Data is replaced with tokens to send to the Cloud and data is encrypted before it leaves the TOE for storage at the Manager data store, ensuring confidentiality in the cloud.
T.KEY_REVEAL	**O.PREVENT_ADMIN_ACCESS.** Information flow control prevents administrators accessing data and keys. **O.ACCESS_CONTROL.** Access controls limit parties that can access application data, such that only users can access data.

Element	Justification
T.TOKEN_EXPLOIT	**O.PREVENT_CLOUD_ACCESS.** The TOE substitutes application data with non-reversible tokens then masks, preventing access to data.
T.RESIDUAL_DATA	**O.PREVENT_CLOUD_ACCESS.** Only masked tokens are sent to the Cloud. **O.TOKENIZE.** Data is replaced with tokens to send to the Cloud and data is encrypted before it leaves the TOE for storage at the Manager data store, ensuring confidentiality in the cloud.
T.FOREIGN	**O.PREVENT_CLOUD_ACCESS.** Only masked tokens are sent to the Cloud. **O.TOKENIZE.** Data is replaced with tokens to send to the Cloud and data is encrypted before it leaves the TOE for storage at the Manager data store, ensuring confidentiality in the cloud.
T.PHYSICAL	**O.PREVENT_CLOUD_ACCESS.** Only masked tokens are sent to the Cloud. **O.TOKENIZE.** Data is replaced with tokens to send to the Cloud and data is encrypted before it leaves the TOE for storage at the Manager data store, ensuring confidentiality in the cloud.
T.INTERCEPT	**O.TOKENIZE.** Data is replaced with tokens to send to the Cloud and data is encrypted before it leaves the TOE for storage at the Manager data store, ensuring confidentiality in the cloud. **O.SECURE_COMMS.** Transfer of user data between the TOE and external entities is through encrypted channels (HTTPS).
T.AUDIT_DATA	**O.PREVENT_ADMIN_ACCESS.** CloudMask administrators are prevented from access to user data. **O.DATA_INTEGRITY.** The TOE associates encrypted user data with digital signatures. **O.SECURE_ADMIN.** Administrator actions are logged. OE.TIMESTAMPS. Environment (Windows 7 and Windows 2008 R2)

Element	Justification
	provides timestamps for audit data.
T.UPDATE_EXPLOIT	**O.VERIFIABLE_UPDATE.** Ability to check integrity of updates prior to installation. **O.SECURE_COMMS.** Secure sessions are established when downloaded updates to the TOE.
OSP.CRYPTO_ENGINE	**OE.CRYPTO_ENGINE.** The environment provides a crypto engine for key and certificate generation and management. **OE.ACCOUNT_LOCKING.** The crypto engine uses digital certificates for account access and controls session security. **OE.ID_AUTH.** Digital certificates are generated and checked by the crypto engine for user authentication.
OSP.DATA_STORE	**OE_DATA_STORE.** User application data and signed policies are stored on the data store, including information required for retrieval.
A.TRUSTED_ADMIN	**OE.MANAGEMENT.** The TOE ensures only trusted administrators (Security Officers) can access the TSF.

Table 12: Suitability of SFRs

Objectives	SFRs
O.PREVENT_ADMIN_ACCESS	**FIA_UAU.2.** Requires a user to be authenticated, preventing non-data owner access. **FIA_UID.2.** Requires a user to be identified, preventing non-data owner access. **FDP_ACC.2.** Access control prevents external entities access to data. **FDP_ACF.1.** Users must have the digital certificate associated with user data to view. **FDP_ETC.2.** Security attributes associated with user data are required for export of data outside the TOE, restricting

Objectives	SFRs
	ability of non-data owners to access data. **FDP_IFF.1.** Element rules, recipient lists and associated digital certificates restrict access to user data to data owner(s). **FDP_UCT.1.** Data is encrypted before it leaves the TOE, such that only data owners authenticated at the TOE have decryption keys.
O.PREVENT_CLOUD_ACCESS	**FIA_UAU.2.** Requires a user to be authenticated, preventing non-data owner access. **FIA_UID.2.** Requires a user to be identified, preventing non-data owner access. **FDP_ACC.2.** Access control prevents external entities access to data. **FDP_ACF.1.** Users must have the digital certificate associated with user data to view/modify. **FDP_ETC.2.** Security attributes associated with user data are required for export of data outside the TOE, restricting ability of non-data owners to access data. **FDP_IFC.1.** Information flow control policies restrict ability of external entities to access user data. **FDP_IFF.1.** Element rules, recipient lists and associated digital certificates restrict access to user data to data owner(s). **FDP_UCT.1.** Data is encrypted before it leaves the TOE, such that only data owners authenticated at the TOE have decryption keys.
O.ACCESS_CONTROL	**FIA_UAU.2.** Requires a user to be authenticated, preventing non-data owner access. **FIA_UID.2.** Requires a user to be identified, preventing non-data owner access. **FDP_ACC.2.** Access control prevents non-authorised entities access to data. **FDP_ACF.1.** Users must have the digital certificate associated with user data to view/modify. **FDP_IFC.1.** Information flow control policies restrict ability of external entities to access user data. **FDP_IFF.1.** Element rules, recipient lists and associated digital certificates restrict access to user data to data owner(s). **FDP_ITC.2.** Only permits users authorised under access control and information flow control policies to import user data. **FMT_MSA.2.** Secure digital certificates enforce access

Objectives	SFRs
	control policies. **FMT_SMR.1.** Access to the TOE is restricted to authorised users. **FPT_TDC.1.** Element rules and decryption requiring keys available to users granted access to data are applied to access user application data.
O.VERIFIABLE_UPDATE	**FMT_MSA.2.** Digital signatures are applied to updates, and the TOE uses these to ensure integrity of updates.
O.TOKENIZE	**FDP_ETC.2.** Encrypted data is associated with a digital signature before export outside the TOE to the Manager data store. **FDP_ITC.2.** Importing data requires the application data access control SFP to be invoked, requiring digital certificate verification for import of user data. **FDP_UCT.1.** Data is encrypted before it leaves the TOE. **FMT_MSA.2.** Security attributes of user data, including digital signatures, are restricted from change. **FPT_ITC.1.** Data transmitted between the TSF and external trusted IT entities is over an encrypted channel (HTTPS).
O.DATA_INTEGRITY	**FAU_GEN.1.** Audit records are generated for purposes of detecting changes to TSF and user application data. **FAU_GEN.2.** Audit records are generated for purposes of detecting changes to TSF and user application data, and are associated with the identity of change. **FMT_MSA.2.** Strong security attributes in the form of encryption/digital certificates/signatures, help to ensure integrity of data. **FPT_ITC.1.** A trusted channel between the TSF and trusted external IT entities provides additional assurance for integrity of data in transit. **FTP_ITC.1.** A secure communications channel is established to ensure integrity of transferred data between the TSF and another trusted IT product.
O.SECURE_COMMS	**FDP_UCT.1.** Data is encrypted before it leaves the TOE. **FPT_ITC.1.** A trusted channel between the TSF and trusted external IT entities provides additional assurance for integrity of data in transit.
O.SECURE_ADMIN	**FAU_GEN.1.** Administrator actions are logged. **FAU_GEN.2.** Administrator actions are associated with identity. **FIA_UAU.2.** Administrators must be authenticated prior to

Objectives	SFRs
	access to the TOE. **FIA_UID.2.** Administrators must be identified prior to access to the TOE. **FMT_MOF.1.** Management of security functions is restricted to the Security Officer. **FMT_MSA.1.** Security attributes can only be modified by the Security Officer. **FMT_MSA.3.**The Security Officer is charged with setting up the TOE in a secure state. **FMT_SMF.1.** Management functions defined are available to the Security Officer. **FMT_SMR.1.** One of the two roles possible at the TOE is that of the Security Officer (TOE administrator).

7.4 Security Requirements Rationale

Table 13: Suitability of SFRs

	O.PREVENT_ADMIN_ACCESS	O.PREVENT_CLOUD_ACCESS	O.ACCESS_CONTROL	O.VERIFIABLE_UPDATE	O.TOKENIZE	O.DATA_INTEGRITY	O.SECURE_COMMS	O.SECURE_ADMIN
FAU_GEN.1						X		X
FAU_GEN.2						X		X
FIA_UAU.2	X	X	X					X
FIA_UID.2	X	X	X					X
FDP_ACC.2	X	X	X					
FDP_ACF.1	X	X	X					
FDP_ETC.2	X	X			X			

	O.PREVENT_ADMIN_ACCESS	O.PREVENT_CLOUD_ACCESS	O.ACCESS_CONTROL	O.VERIFIABLE_UPDATE	O.TOKENIZE	O.DATA_INTEGRITY	O.SECURE_COMMS	O.SECURE_ADMIN
FDP_IFC.1		X	X					
FDP_IFF.1	X	X	X					
FDP_ITC.2			X		X			
FDP_UCT.1	X	X			X		X	
FMT_MOF.1								X
FMT_MSA.1								X
FMT_MSA.2			X	X	X	X		
FMT_MSA.3								X
FMT_SMF.1								X
FMT_SMR.1			X					X
FPT_ITC.1					X	X	X	
FPT_TDC.1			X					
FTP_ITC.1						X		

7.5 TOE Summary Specification Rationale

70 Table 14 provides a coverage mapping showing that all SFRs are mapped to the security functions described in the TSS.

Table 14: Map of SFRs to TSS Security Functions

SFR	Data Handling Rules	Data Tokenization	Access Control Policies	Protected Communications	Information Flow Control	Trusted Update
FAU_GEN.1			X			X
FAU_GEN.2			X			X
FIA_UAU.2			X		X	X
FIA_UID.2			X		X	X
FDP_ACC.2	X		X		X	
FDP_ACF.1	X		X		X	
FDP_ETC.2	X	X	X		X	
FDP_IFC.1					X	
FDP_IFF.1	X		X		X	
FDP_ITC.2	X		X		X	
FDP_UCT.1	X	X		X	X	
FMT_MOF.1	X					X
FMT_MSA.1	X					X
FMT_MSA.2		X				X
FMT_MSA.3	X				X	
FMT_SMF.1	X				X	X
FMT_SMR.1	X					
FPT_ITC.1				X		
FPT_TDC.1	X	X				

SFR	Data Handling Rules	Data Tokenization	Access Control Policies	Protected Communications	Information Flow Control	Trusted Update
FTP_ITC.1				X	X	X

7.6 Security Requirements Dependency Analysis

Table 15: SFR Dependency Analysis

Security Functional Requirement	Dependencies	Resolution
FAU_GEN.1	FPT_STM.1 Reliable time stamps	Not Fulfilled. The TOE timestamps audit records, and relies on the Windows 7 and Windows 2008 R2 real-time timestamps for timestamping audit data. See OE.TIMESTAMPS.
FAU_GEN.2	FAU_GEN.1 Audit data generation FIA_UID.1 Timing of identification	FAU_GEN.1 FIA_UID.2 (Hierarchical to FIA_UID.1)
FIA_UAU.2	FIA_UID.1 Timing of identification	FIA_UID.2 (Hierarchical to FIA_UID.1)
FIA_UID.2	No dependencies	
FDP_ACC.2	FDP_ACF.1 Security attribute based access control	FDP_ACF.1
FDP_ACF.1	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	FDP_ACC.2 (Hierarchical to FDP_ACC.1) FMT_MSA.3

Security Functional Requirement	Dependencies	Resolution
FDP_ETC.2	FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control	FDP_ACC.2 (Hierarchical to FDP_ACC.1) and FDP_IFC.1
FDP_IFC.1	FDP_IFF.1 Simple security attributes	FDP_IFF.1
FDP_IFF.1	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation	FDP_IFC.1 FMT_MSA.3
FDP_ITC.2	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path] FPT_TDC.1 Inter-TSF basic TSF data consistency	FDP_ACC.2 (Hierarchical to FDP_ACC.1) and FDP_IFC.1 FTP_ITC.1 FPT_TDC.1
FDP_UCT.1	[FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path] [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	FTP_ITC.1 FDP_ACC.2 (Hierarchical to FDP_ACC.1) and FDP_IFC.1

Security Functional Requirement	Dependencies	Resolution
FMT_MOF.1	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FMT_SMR.1 FMT_SMF.1
FMT_MSA.1	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FDP_ACC.2 (Hierarchical to FDP_ACC.1) and FDP_IFC.1 FMT_SMR.1 FMT_SMF.1
FMT_MSA.2	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FDP_ACC.2 (Hierarchical to FDP_ACC.1) and FDP_IFC.1 FMT_MSA.1 FMT_SMR.1
FMT_MSA.3	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FMT_MSA.1 FMT_SMR.1
FMT_SMF.1	No dependencies	
FMT_SMR.1	FIA_UID.1 Timing of identification	FIA_UID.2 (Hierarchical to FIA_UID.1)
FPT_ITC.1	No dependencies	
FPT_TDC.1	No dependencies	
FTP_ITC.1	No dependencies	