MagicDBPlus v2.0 Security Target

Version 1.3

< **Change History >**

Version	Date	Contents / Prepared by
1.0	Dec 04, 2017	Formulated / Dreams Security, Infrasolution Team 1
1.1	Mar 03, 2020	Modified / Dreamsecurity, Security Technology Lab, R&D Part 1 R&D Team 2 : Changed the organization of contents
1.2	Apr 10, 2020	Modified / Dream Security, Security Technology Lab, R&D Part 1 R&D Team 2: Modified the extended components definition and definition of security requirements
1.3	July 10, 2020	Modified / Dream Security, Security Technology Lab, R&D Part 1 R&D Team 2: Modified TOE version

< Table of Contents >

1 Security Target Introduction

1.1 Security Target Reference

Cls.	Contents
Title	MagicDBPlus v2.0 Security Target
ST Version	v1.3
Prepared by	Dreamsecurity Co.,Ltd, R&D-Part 1 R&D-Team 2
Date Prepared	July 07, 2020
Evaluation Criteria	Common Criteria for Information Technology Security Evaluation (Ministry of Science, ICT and Future Planning Notice No. 2013-51)
Common Criteria Version	V3.1 r5
Evaluation Assurance Level	EAL1+(ATE_FUN.1)
Keywords	Database (DB), DBMS, encryption, decryption, Oracle

1.2 TOE Reference

Cls.		Contents
TOE		MagicDBPlus v2.0
Version		v2.0.3.0
TOE Component	Management Server	MagicDBPlus v2.0 Server v2.0.3.0 : MagicDBPlus_v2.0_Server_v2.0.3.0.sh
	Administrative Tool	MagicDBPlus v2.0 Admin v2.0.3.0 : MagicDBPlus_v2.0_Admin_v2.0.3.0.exe
	Agent	MagicDBPlus v2.0 Agent v2.0.3.0 : MagicDBPlus_v2.0_Agent_v2.0.3.0.sh
Guidance Document	Installation Guide	MagicDBPlus v2.0 Installation Guide v1.2 : MagicDBPlus_v2.0_PRE_v1.2.pdf
	Operational Guidance	MagicDBPlus v2.0 Operational Guidance v1.2 : MagicDBPlus_v2.0_OPE_v1.2.pdf
Developer		Dreamsecurity Co.,Ltd, R&D-Part 1 R&D-Team 2

1.3 TOE Overview

1.3.1 Overview of Database Encryption

MagicDBPlus v2.0 (hereinafter referred to as “TOE”) plays a role in encrypting database (“DB”) and preventing unauthorized disclosure of information to be protected.

Subjected to encryption of the TOE, the DB is controlled by the Database Management System (“DBMS”) in an organization’s operational environment. In this ST, all the data before and after being encrypted and stored in the DB are defined as user data. According to security policy of the organization that operates the TOE, the whole or part of the user data could be subject to encryption.

1.3.2 TOE Type and Scope

The TOE is provided in a form of software and offers a function of encrypting/decrypting user data as per column. Types of the TOE defined herein include a “Plug-in” under which the TOE is comprised of an Agent, a Management Server and an Administrative Tool.

TOE components are shown in the following [Table 1-1].

[Table 1-1] TOE Components

Component	Contents
MagicDBPlus v2.0 Server v2.0.3.0	Serves as storage of cryptographic key management and audit data for encryption/decryption of the TOE’s database
MagicDBPlus v2.0 Admin v2.0.3.0	Provides establishment and management of encryption/decryption policies through a console, in order to control the TOE
MagicDBPlus v2.0 Agent	Conducts encryption/decryption of user DB data with a plug-in installed to

v2.0.3.0	the database server that has DB under the protection of the TOE

DEKs (Data Encryption Key) used to encrypt/decrypt user data of the TOE are encrypted into KEKs (Key Encryption Key) for protection. In addition, DEKs are also used for protection of communications between TSF data stored and TOE components and utilize a validated cryptographic module whose security and implementation conformity have been verified through the Korea Cryptographic Module Validation Program (KCMVP).

- Cryptographic Module Name : MagicCrypto V2.2.0
- Validation No : CM-162-2025.3
- Validation Date : 2020-03-03

1.3.3 TOE Usage and Major Security Features

Along with an Agent installed in the database server that has the DB to be protected, the TOE encrypts user data received from the application server before storing them into the DB and decrypts the user data encrypted which have been transmitted from the database server to the application server in accordance with policies defined by the authorized administrator who performs encryption/decryption of user data as prescribed by the scope of encryption objects required for an organization's security policy via the Management Server, using the Administrative Tool. The TOE executable codes offer integrity.

In order for the authorized administrator to operate the TOE under an organization's operational environment in a secure manner, it provides security audit function that records and manages audit data regarding major auditable events; management of cryptographic keys for user and TSF data encryption; cryptographic support such as cryptographic operation; user data protection that encrypts user data and safeguards residual information; verification of the authorized administrator's identity and authentication failure handling; identification and authentication such as TOE internal mutual authentication; definition of security functions and roles; security management functions for configuration; protection of TSF data transmitted among TOE components; protection of TSF data stored in repositories controlled by the TSF; TSF protection function such as TSF self-testing; and provision of TOE access to manage access session(s) of the authorized administrator; and the Agent performs encryption/decryption of user data in accordance with the security policy defined.

1.3.4 Non-TOE and TOE Operational Environment

The TOE is software that provides a function of preventing unauthorized disclosure of information that intends to be protected with encryption of the DB. All hardware and operating system (OS) where the TOE is installed and DBMS are regarded as non-TOE. As an external IT entity, the mail server (SMTP , SMTP Server) is used to notify an alarm to an administrator with respect to threats that occur in operation.

Hardware/software required for the TOE to be installed is listed in [Table 1.2].

[Table 1-2] Installation Hardware/Software installed for TOE

TOE	Category	Item	Minimum Specifications
Agent	H/W	CPU	Intel Core i5 CPU 2.30 GHz or higher
		Memory	4 GB or more
		HDD	500 MB or more of space required for TOE installation
		NIC	Ethernet 100/1000 Mbps * 1 Port or more
	S/W	OS	CentOS 7.8 (Linux Kernel 3.10.0) 64 bit
		DBMS to be protected	Oracle 12.2.0.1.0 64 bit
Management Server	H/W	CPU	Intel Core i3 CPU 2.27 GHz or higher
		Memory	4 GB or more
		HDD	100 GB or more of space required for TOE installation
		NIC	Ethernet 100/1000 Mbps * 1 Port or more
	S/W	OS	CentOS 7.8 (Linux Kernel 3.10.0) 64 bit
Administrative Tool	H/W	CPU	Intel Core i5 CPU 2.50 GHz or higher
		Memory	4 GB or higher
		HDD	500 MB or more of space required for TOE installation
		NIC	Ethernet 100/1000 Mbps * 1 Port or more

	S/W	OS	Windows 10 Pro 64 bit

Furthermore, the software required for the TOE is as shown below.

- The following is the external IT entity required for the TOE.

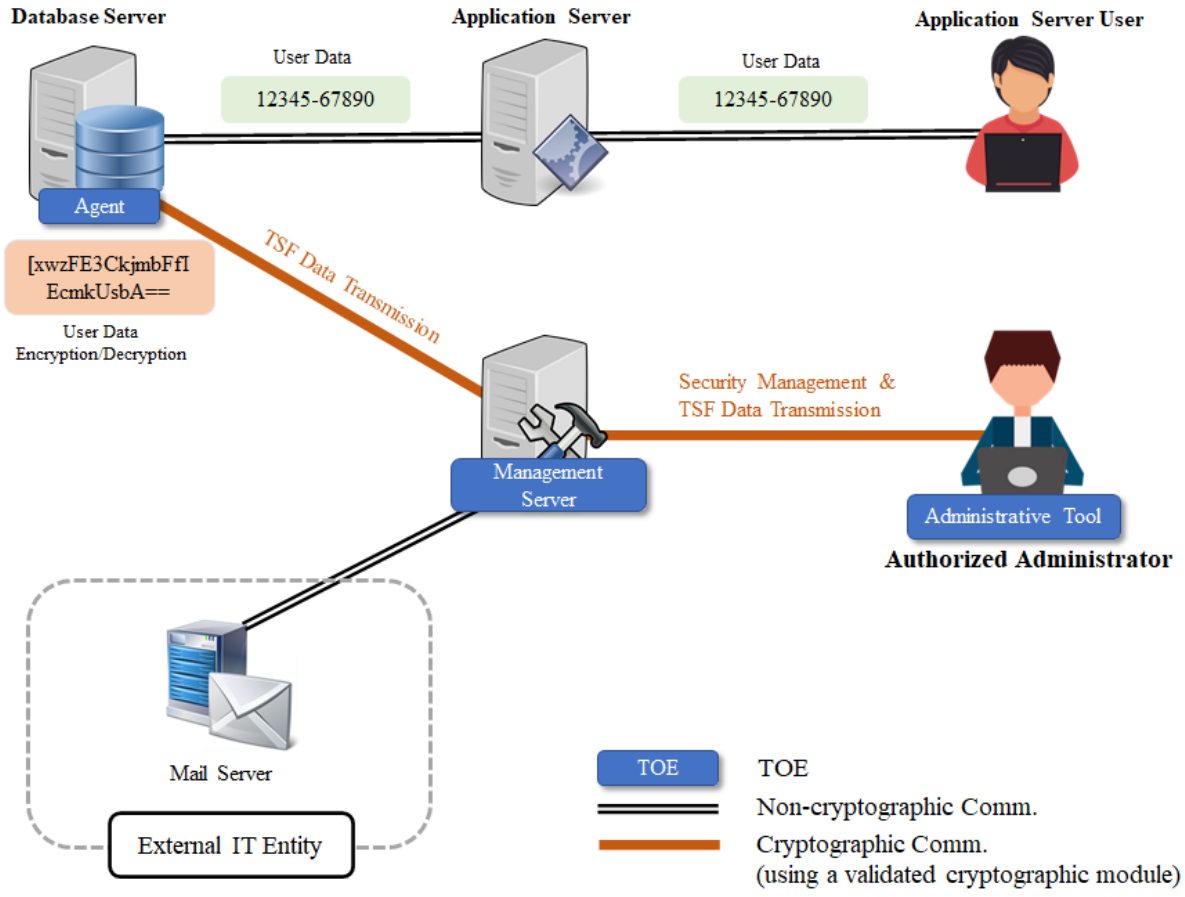
External IT Entity	Description
Mail Server	To be used to notify or send an alarm (a warning mail) to an administrator regarding threats that arise during operation of the Management Server

- The following is the 3rd party software included in the TOE.

3rd party SW	Description
SQLite v3.32.3	To be used as DB for repository files of TSF data (DEKs for user data, critical security parameters, TOE configuration values and audit data, etc.) in the Management Server

1.3.5 TOE Operational Environment

As shown in Figure 1-1, the operational environment is comprised of MagicDBPlus v2.0 Agent ("Agent"), MagicDBPlus v2.0 Server ("Management Server") and MagicDBPlus v2.0 Admin ("Administrative Tool").
Installed as a plug-in to the DBMS which has to be protected, the Agent receives TSF data from the Management Server and performs encryption/decryption of user data upon the request from the application server. In addition, the authorized administrator manages the scope and policies of encryption befitting security policy required in the organization via the Management Server, using the Administrative Tool. Upon the request of application service users, the application server makes a request to the database server while the Agent encrypting/decrypting user data, if necessary, and deliver them to application service users.
Moreover, if a critical event (e.g., reaching to audit data threshold, etc.) arises in the Management Server, a mail is sent to a user designated by the authorized administrator via a mail server.



[Figure 1-1] TOE Operational Environment

Furthermore, communications among TOE components, which rely on a self-implemented protocol, carry out cryptographic communication, using an approved algorithm of the validated cryptographic module (MagicCrypto V2.2.0).

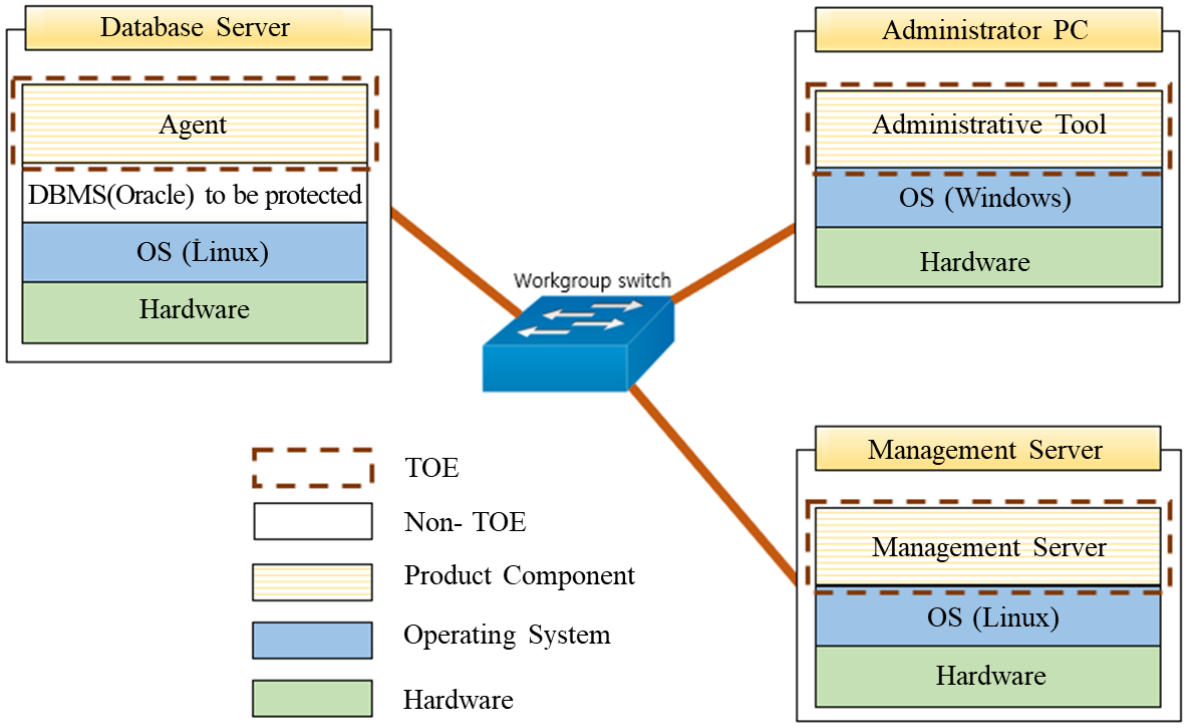
1.4 TOE Description

This Chapter describes the physical and logical scopes of the TOE.

1.4.1 Physical Scope of the TOE

The TOE consists of the Management Server, the Administrative Tool, the Agent, an operational guidance (MagicDBPlus v2.0 Operational Guidance v1.2, MagicDBPlus_ v2.0_OPE_v1.2.pdf) and an installation guide (MagicDBPlus v2.0 Installation Guide v1.2, MagicDBPlus_v2.0_PRE_v1.2.pdf).

The Management Server is MagicDBPlus v2.0 Server v2.0.3.0 (MagicDBPlus_v2.0_Server_v2.0.3.0) that plays a role in login processing, policy establishment, audit data storage and generation and management of encryption keys.



[Figure 1-2] Physical Scope of the TOE

The Administrative Tool refers to MagicDBPlus v2.0 Admin v2.0.3.0 (MagicDBPlus_v2.0_Admin_v2.0.3.0) which is software offering the security management function to the administrator whereas the Agent refers to MagicDBPlus v2.0 Agent v2.0.3.0 (MagicDBPlus_v2.0_Agent _v2.0.3.0) which is software that provides a function of encrypting/decrypting user data. Hardware and OS where the TOE is installed are not included in the scope of the TOE.

The following are types and delivery methods as per TOE component.

Category		Type	Delivery
TOE Component	MagicDBPlus v2.0 Server v2.0.3.0 : MagicDBPlus_v2.0_Server_v2.0.3.0.sh	S/W	Included in an installation CD of the product package provided to users
	MagicDBPlus v2.0 Admin v2.0.3.0 : MagicDBPlus_v2.0_Admin_v2.0.3.0.exe	S/W	

	MagicDBPlus v2.0 Agent v2.0.3.0 : MagicDBPlus_v2.0_Agent_v2.0.3.0.sh	S/W	
Guidance Document	MagicDBPlus v2.0 Installation Guide v1.2 : MagicDBPlus_v2.0_PRE_v1.2.pdf		
	MagicDBPlus v2.0 Operational Guidance v1.2 : MagicDBPlus_v2.0_OPE_v1.2.pdf		

1.4.2 Logical Scope of the TOE

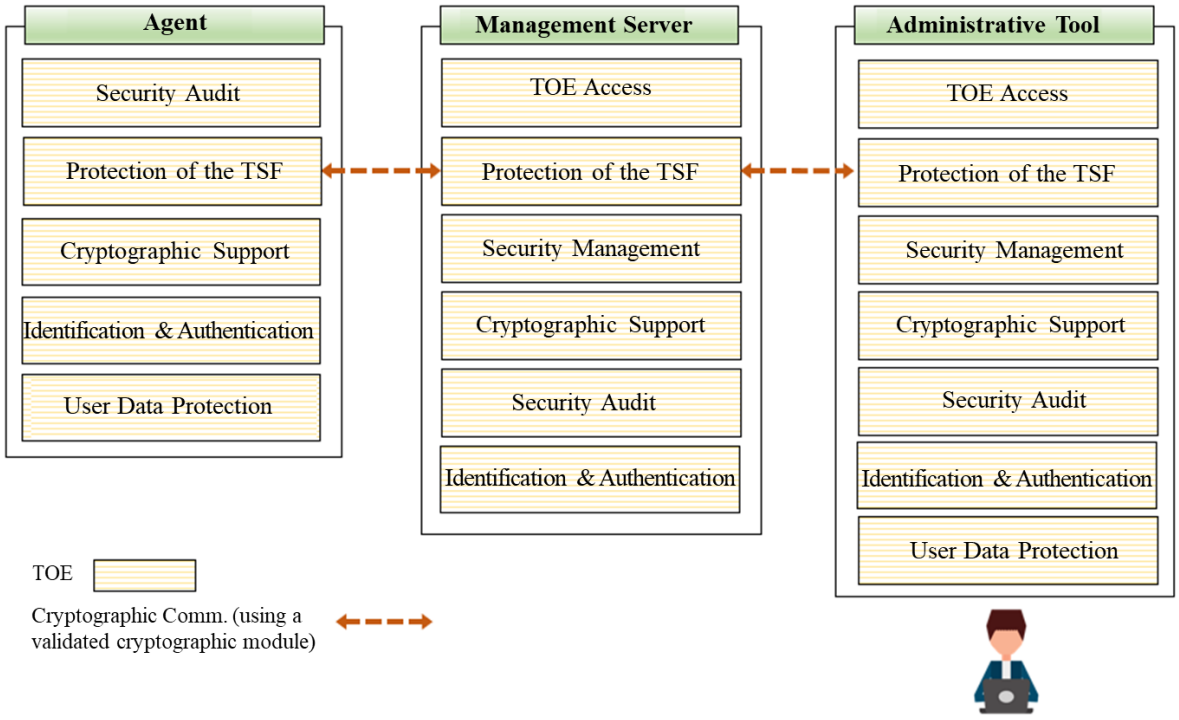


Figure 1-3 Logical Scope of the TOE

Security Audit

The TOE records the data/time of an event, an event type, subject identity and authority, an outcome of an event and event details as audit data which are stored and controlled in the Management Server. The audit data generated can be viewed by an administrator, using the Administrative Tool called MagicDBPlus v2.0 Admin. There is only one super administrator who is capable of viewing security management and audit records as an authorized administrator to which an alarm is send via mail in case of any access from an unauthorized user.

In addition, the outcome of a self-test administered at each component of the TOE is stored and controlled in the Management Server as well. In case a self-test fails, an alarm is sent to an email defined by the authorized administrator.

In case the audit data storage reaches a certain threshold defined by the administrator, a warning email is sent to the administrator. Also, if the audit storage is full, audited obsolete events are overwritten and a warning message is sent to the administrator via email.

In case audit data stored exceed the capacity assigned by the administrator, a warning message is sent to the administrator via email. Also, audit data storage is full, obsolete audited event data are overwritten and a warning message is sent to the administrator as well.

All audit data are encrypted and stored in the local file DB of the Management Server, with a limited access only from approved Management Server processes.

Cryptographic Support

The TOE performs generation, distribution, destruction and operation of cryptographic keys and random bit generation, using a validated cryptographic module, Magic Crypto V2.2.0. The cryptographic module is also used to generate and exchange cryptographic keys during cryptographic communications among TOE components physically separated.

The TOE is generate the cryptographic key is generated, a cryptographic key is generated using a random bit through a random bit generator (HASH_DRBG 256) of the validated cryptographic module, and to encrypt user data of the DBMS to be protected is encrypted, a symmetric key encryption algorithm (ARIA-CBC 128/192/256 bit, SEED-CBC 128 bit, LEA-CBC 128/192/256 bit) and hash algorithm(SHA-256/384/512).

In addition, a symmetric key encryption algorittm (ARIA-CBC 256 bit), digital signature algorittm (RSA-PSS 2048 bit), hash algorithm (SHA-256) and MAC algorithm(HMAC-SHA 256, 1024 bit) are used for protction of TSF data.

Cryptograchic key distribution between the TOE components is safely distributed through public key encryption method (RSAES 2048 bit), and the cryptographic key is overwritten with "0x00" for destruction.

User Data Protection
The TOE conducts encryption/decryption in case of storage and modification of user data in DB to be encrypted, using the validated cryptographic module, MagicCrypto V2.2.0, in accordance with encryption/decryption policy for user data defined by the authorized administrator.
Operated as a plug-in, the TOE supports encryption/decryption of user data as per column in the Agent.
The TOE generates different encryption values each time it conducts encryption of the same user data.
Once the encryption/decryption is completed, it performs initialization not to restore the previous value of the original user data.

Identification and Authentication
The TOE provides the identification and authentication function to an administrator who carries out security management function so that the ID and password have to be changed during the initial login after installation of the product. When the identification and authentication data of an administrator are entered, the password is masked with “*” to protect the authentication feedback.
Moreover, in case of an authentication failure, feedback on a failure reason is not provided, and the account become locked (for five minutes) after consecutive authentication failures (5 times).
It also blocks an attempt to re-use authentication information regarding the administrator who has logged in to the TOE.
The TOE provides the following criteria with respect to verification of the password:

- Length of password: (Min.) 9 characters ~ (Max.) 16 characters
- Characters usable for the password: English upper and low cases, numbers and special characters (~, `, !, @, #, $, %, ^, &, *, (,), -, _, +, =)
- The password has to be a combination of the following three: English upper/lower cases, numbers and special characters.

The TOE carries out mutual authentication, using a self-implemented protocol for secure communications among TOE components.

Security Management
There is only one account for an authorized administrator in the TOE and during the initial login, the ID and password have to be changed.
The TOE provides security management functions including generation and deletion of cryptographic keys, registration and deletion of policies, mail notification setting, audit threshold setting and encryption/decryption of user data. The authorized administrator performs security management, using a security management interface available in the Administrative Tool.

Protection of the TSF
The TOE protects TSF data stored in repositories controlled by the TSF and those transmitted among TOE components and conducts inspection on major security function processes based on TSF self-tests. The TOE runs a self-test on the main process, major files and the cryptographic module during the initial start-up of the TOE and normal operation on a periodical basis (3 hours after start-up). In terms of major files, the authorized administrator can manually run such a self-test via a security management screen. In case the self-test shows any abnormal behavior, a warning mail is sent to the administrator.

TOE Access
The number of management access sessions available for the administrator to enforce security management functions is limited to one. If there is an administrator session already logged in to the Management Server, no

more authorized administrator can access.
In case no action is detected for a certain period of time (default value: 10 minutes) after the authorized administrator has logged in to the Management Server via the Administrative Tool, the session accessible to the Management Server is closed.
In addition, the authorized administrator can have no more than two IPs, and during the initial installation of the Management Server, one IP accessible during the installation process has to be pre-defined.

1.5 Terms and Definitions

Application Server
Application server defined in this ST means a server in which an application developed for provision of specific application services is installed and operated in an organization that runs a TOE. The corresponding application plays a role in reading user data from DB that exists in the database server upon the request of an application service user or in transmitting the user data to be stored in DB to the database server.

Approved Cryptographic Algorithm
Cryptographic algorithm selected by the Cryptographic Module Validation Authority for a block cipher, hash function, massage authentication code, random bit generator, key setting, public key encryption and digital signature cryptographic algorithm, in consideration of security, reliability and interoperability.

Approved Mode of Operation
Mode of a cryptographic module that uses an approved cryptographic algorithm.

Assets
Entities that the owner of the TOE presumably places value upon.

Assignment
The specification of an identified parameter in a component (of the CC) or requirement.

Attack Potential
Measure of the effort to be expended in attacking a TOE, expressed in terms of an attacker's expertise, resources and motivation.

Augmentation
Addition of one or more requirement(s) to a package.

Authentication Data
Information used to verify the claimed identity of a user.

Authorized Administrator
Authorized user who operates and manages the TOE in a secure manner.

Authorized User
TOE user who may, in accordance with the SFRs (Security Function Requirements), perform an operation.

Can/could
"Can" or "could" presented in Application notes indicates an optional requirement that may be applied to the TOE depending on a choice of ST author.

Class
Set of CC families that share a common security purpose.

Column
Set of data values with a specific data type which corresponds to a value of each row in the relational datable table.

Component
Smallest selectable set of elements on which requirements may be based.

Critical Security Parameters (CSP)
Security-related information (e.g., a secret key/personal key and authentication data such as a password or personal ID number), if disclosed or modified, that could comprise the security of a cryptographic module.

Data Encryption Key (DEK)
Key that encrypting/decrypting data.

Database (DB)
Collection of data compiled in accordance with a certain structure to accept, store and provide data in response to needs of multiple users, in order to support multiple applications in parallel / The database related to encryption as per column required in this protection profile (PP) also refers to a relational database.

Database Management System (DBMS)
Software system established to configure and apply a database / The DBMS related to encryption as per column required in this PP refers to a database management system based on the relational database model.

Decryption
Restoration of a cryptogram into original plain texts using a decryption key.

Dependency
Relationship between components such that if a requirement based on the depending component is included in a PP, ST or package, a requirement based on the component that is depended upon must normally also be included in the PP, ST or package.

Element
Minimum unit of an indivisible security need (requirement)

Encryption
Conversion of plain texts into a cryptogram using an encryption key.

Evaluation Assurance Level (EAL)
Set of assurance requirements drawn from CC Part 3, representing a point on the CC predefined assurance scale, that form an assurance package.

External Entity
Human or IT entity possibly interacting with the TOE from outside of the TOE boundary.

Family
Set of components that share a similar goal but differ in emphasis or rigour.

Identity
Unique representation to identify an authorized user / It could be the real name, alias or nick name of the user.

Iteration
Use of the same component to express two or more distinct requirements.

Key Encryption Key (KEK)
Key that encrypting/decrypting other cryptographic keys.

Management Access
Act of the administrator attempting access for the purpose of the TOE management, using HTTPS, SSH and TLS, etc.

Management Console
Application program such as GUI (Graphical User Interface) or CLI (Command Line Interface) provided to an administrator for management and configuration of a system / It is also used as a synonym with the Administrative Tool in this document.

Nonce
Cryptographic token arbitrarily generated to prevent a replay attack.

Object
Passive entity in the TOE, that contains or receives information, and upon which subjects perform operations.

Operation (on a component of the CC)
Modification or repetition of a component / Allowed operations on components are assignment, iteration, refinement and selection.

Operation (on a subject)
Specific type of action performed by a subject on an object.

Organizational Security Policies
Set of security rules, procedures or guidelines currently assigned or regarded to be assigned in the operational environment by a real or virtual organization.

Private Key
A cryptographic key which is used in an asymmetric cryptographic algorithm and is uniquely associated with an entity(the subject using the private key), not to be disclosed

Protection Profile (PP)
Implantation-independent statement of security needs for a TOE type.

Public Key
A cryptographic key which is used in an asymmetric cryptographic algorithm and is associated with an unique entity(the subject using the public key), it can be disclosed

Public Key(asymmetric) cryptographic algorithm
A cryptographic algorithm that uses a pair of public and private key

Public Security Parameters (PSP)
Security-related public information whose modification could compromise the security of a cryptographic module.

Random Bit Generator (RBG)
Device or algorithm that outputs a binary sequence that is statistically independent and unbiased / A random bit generator used for cryptographic application normally generates a bit string of 0 and 1, and such a sequence can be combined into a random bit block. The RBG is classified into deterministic and non-deterministic. The deterministic RBG is comprised of an algorithm that generates a bit string from the initial value called a seed key whereas the non-deterministic RBG produces output that depends on unpredictable physical sources.

Recommend/be recommended
"Recommend" or "be recommended" presented in Application notes is not a mandatory requirement applied to the TOE but a requirement encouraged to be applied for secure operation of the TOE.

Refinement
Addition of details to a component.

Role
Predefined set of rules on permissible interactions between a user and the TOE.

Secret Key
The cryptographic key which is used in symmetric cryptographic algorithm and is associated with on or more entity, it is not allowed to release

Secure Sockets Layer (SSL)

Security protocol suggested by Netscape to provide security such as confidentiality and integrity in a computer network.

Security Attribute
Property of subjects, users (including external IT products), objects, information, sessions and/or resources that is used in defining the SFRs and whose values are used in enforcing the SFRs.

Security Function Policy (SFP)
Set of rules describing specific security behavior enforced by the TSF (TOE Security Functionality) and are expressible as a set of SFRs (Security Function Requirements).

Security Policy Document
Document uploaded to the list of validated cryptographic modules along with module names / It is a summary document that specifies types of cytoproct modules, validated cryptographic algorithms provided by cryptographic modules and operational environments.

Security Target (ST)
Implementation-dependent statement of security needs for a specific identified TOE.

Selection
Specification of one or more items from a list in a component.

Self-test
Pre-operational or conditional test executed by a cryptographic module.

Shall/must
"Shall" or "must" presented in Application notes indicates a mandatory requirement applied to the TOE.
Subject Active entity in the TOE that performs operations on objects.

Symmetric Cryptographic Technique
Encryption technique that uses the same secret key in the modes of encryption and decryption / It is also known as a secret key cryptographic technique.

Target of Evaluation (TOE)
Set of software, firmware and/or hardware possibly accompanied by guidance.

Threat Agent
Unauthorized external entity that causes a threat such as unlawful access to, change or deletion of an asset .

TOE Security Functionality (TSF)
Combined functionality of all hardware, software, and firmware of a TOE that must be relied upon for the correct enforcement of the SFRs.

Transport Layer Security (TLS)
Cryptographic authentication communication protocol between a SSL-based server and a client, that is specified in RFC 2246.

TSF Data
Data generated by and for the TOE that could affect operation of the TOE.

User
See "external entity"

User Data
Data for users without any influence on the TSF (TOE Security Functionality)

Validated Cryptographic Module
Cryptographic module that is validated/approved and given a validation number by the Cryptographic Module

Validation Authority

1.6 Conventions

In this ST, English is also used for some acronyms and better delivery of meaning. The notation, formatting and conventions are consistent with the Common Criteria for Information Technology Security Evaluation.

The CC allows several operations to be performed for security functional requirements: iteration, assignment, selection and refinement.

Each operation is used in this ST.

Iteration
Iteration is used when a component is repeated with varying operations. The result of iteration is marked with an iteration number in parenthesis following the component identifier, i.e., denoted as (iteration No.).

Assignment
This is used to assign specific values to unspecified parameters (e.g., password length). The result of assignment is indicated in square brackets, i.e., [assignment value].

Selection
This is used to select one or more options provided by the CC in stating a requirement. The result of selection is shown as *underlined and italicized*.

Refinement
This is used to add details and thus further restrict a requirement. The result of refinement is shown in **bold text**.

2 Conformance Claim

2.1 CC, PP and Security Requirements Package Conformance

The Common Criteria (CC) and the Protection Profile (PP) and the Security Requirements Package to which this ST and the TOE conform are as follows:

Category	Conformance
CC	Common Criteria for Information Technology Security Evaluation, Version 3.1, Rev. 5 - Common Criteria for Information Technology Security Evaluation. Part 1: Introduction and General Model, Version 3.1r5 (CCMB-2017-04-001, April, 2017) - Common Criteria for Information Technology Security Evaluation. Part 2: Security Functional Components, Version 3.1r5 (CCMB-2017-04-002, April, 2017) - Common Criteria for Information Technology Security Evaluation. Part 3: Security Assurance Components, Version 3.1r5 (CCMB-2017-04-003, April, 2017)
CC Part 2	Extended: FCS_RBG.1, FIA_IMA.1, FDP_UDE.1, FMT_PWD.1, FPT_PST.1, FTA_SSL.5
CC Part 3	Conformance
PP	Korean National Protection Profile for Database Encryption V1.1 (2019-12-11)
Security Requirements Package	Conformance to EAL1 augmented (ATE_FUN.1)

2.2 Conformance Claim Rationale

This ST strictly conforms to the “Korean National Protection Profile for Database Encryption V1.1,” adopting the TOE types, security objectives and security requirements in the same way as the PP.

Category	PP	ST	Claim Rationale
TOE Type	Classified into ‘Plug-in’ and ‘API’	Plug-in	more restrictive than the PP - In the PP, it is classified into “Plug-in” and “API” whereas the TOE identifies only ‘plug-in’ in the ST, which is more restrictive than PP
Security Objective	OE.PHYSICAL_CONTROL	OE. PHYSICAL_CONTROL	same as the PP
	OE.TRUSTED_ADMIN	OE. TRUSTED_ADMIN	same as the PP

Category	PP	ST	Claim Rationale
	OE.SECURE_DEVELOPMENT	OE. SECURE_DEVELOPMENT	same as the PP
	OE.LOG_BACKUP	OE.LOG_BACKUP	same as the PP
	OE.OPERATION_SYSTEM_REINFORCEMENT	OE. OPERATION_SYSTEM_REINFORCEMENT	same as the PP
	-	OE.TIMESTAMP	more restrictive than the PP - In the PP, there is no security problem definition nor security requirements regarding timestamp used in audit records, but the ST identifies an additional assumption that a secure timestamp is received from the TOE operational environment and is used, which makes it more restrictive than the PP.
Security Functional Requirement	FAU_ARP.1	FAU_ARP.1	same as the PP
	FAU_GEN.1	FAU_GEN.1	same as the PP
	FAU_SAA.1	FAU_SAA.1	same as the PP
	FAU_SAR.1	FAU_SAR.1	same as the PP
	FAU_SAR.3	FAU_SAR.3	same as the PP
	FAU_SEL.1	-	not mandatorily implemented in TOE as an optional SFR
	FAU_STG.1	FAU_STG.1	same as the PP
	FAU_STG.3	FAU_STG.3	same as the

Category	PP	ST	Claim Rationale
			PP
	FAU_STG.4	FAU_STG.4	same as the PP
	FCS_CKM.1(1)	FCS_CKM.1(1)	same as the PP
	FCS_CKM.1(2)	FCS_CKM.1(2)	same as the PP
	FCS_CKM.2	FCS_CKM.2	same as the PP
	FCS_CKM.4	FCS_CKM.4	same as the PP
	FCS_COP.1(1)	FCS_COP.1(1)	same as the PP
	FCS_COP.1(2)	FCS_COP.1(2)	same as the PP
	FCS_RBG.1	FCS_RBG.1	same as the PP
	FDP_UDE.1	FDP_UDE.1	same as the PP
	FDP_RIP.1	FDP_RIP.1	same as the PP
	FIA_AFL.1	FIA_AFL.1	same as the PP
	FIA_IMA.1	FIA_IMA.1(1)	conform to a requirement in the PP for mutual authentication of the section between the Management Server and Agent and of the section between the Management Server and Administrative Tool that are cryptographically communicated among TOE components as required by the PP
		FIA_IMA.1(2)	
	FIA_SOS.1	FIA_SOS.1	same as the PP
	FIA_UAU.1	FIA_UAU.1	same as the PP
	FIA_UAU.4	FIA_UAU.4	same as the PP

Category	PP	ST	Claim Rationale
	FIA_UAU.7	FIA_UAU.7	same as the PP
	FIA_UID.1	FIA_UID.1	same as the PP
	FMT_MOF.1	FMT_MOF.1	same as the PP
	FMT_MTD.1	FMT_MTD.1	same as the PP
	FMT_PWD.1	FMT_PWD.1	same as the PP
	FMT_SMF.1	FMT_SMF.1	same as the PP
	FMT_SMR.1	FMT_SMR.1	same as the PP
	FPT_ITT.1	FPT_ITT.1	same as the PP
	FPT_PST.1	FPT_PST.1	same as the PP
	FPT_STM.1	-	not mandatorily implemented in TOE as an optional SFR
	FPT_TEE.1	-	not mandatorily implemented in TOE as an optional SFR
	FPT_TST.1	FPT_TST.1	same as the PP
	FTA_MCS.2	FTA_MCS.2	same as the PP
	FTA_SSL.5	FTA_SSL.5	same as the PP
	FTA_TSE.1	FTA_TSE.1	same as the PP
	FTP_ITC.1	-	not mandatorily implemented in TOE as an optional SFR
	FTP_TRP.1	-	not mandatorily implemented in TOE as an optional SFR
Assurance Requirement	ASE_INT.1	ASE_INT.1	same as the PP
	ASE_CCL.1	ASE_CCL.1	same as the PP
	ASE_OBJ.1	ASE_OBJ.1	same as the PP
	ASE_ECD.1	ASE_ECD.1	same as the PP
	ASE_REQ.1	ASE_REQ.1	same as the

Category	PP	ST	Claim Rationale
			PP
	ASE_TSS.1	ASE_TSS.1	same as the PP
	ADV_FSP.1	ADV_FSP.1	same as the PP
	AGD_OPE.1	AGD_OPE.1	same as the PP
	AGD_PRE.1	AGD_PRE.1	same as the PP
	ALC_CMC.1	ALC_CMC.1	same as the PP
	ALC_CMS.1	ALC_CMS.1	same as the PP
	ATE_FUN.1	ATE_FUN.1	same as the PP
	ATE_IND.1	ATE_IND.1	same as the PP
	AVA_VAN.1	AVA_VAN.1	same as the PP

3 Security Objectives

3.1 Security Objectives for Operational Environment

The followings are the security objectives handled by technical and procedural methods supported in the operational environment, in order to ensure that the TOE provides security functionality in an accurate manner.

OE.PHYSICAL_CONTROL

The place where the Agent, Management Server and Administrative Tool out of the TOE components are installed and operated shall be equipped with access control and protection facilities so that only authorized administrator can access.

OE.TRUSTED_ADMIN

The authorized administrator of the TOE shall be non-malicious, be appropriately trained for the TOE management functions and accurately perform his or her duties in accordance with the administrator guidelines.

OE.LOG_BACKUP

The authorized administrator of the TOE shall periodically check spare space of audit data storage in case of any loss of audit records and perform backup (to an external log server or separate storage device, etc.) to prevent audit data loss.

OE.OPERATION_SYSTEM_REINFORCEMENT

The authorized administrator of the TOE shall ensure reliability and security of the operation system by performing reinforcement to reduce latest vulnerabilities of the operating system in which the TOE is installed and operated.

OE.SECURE_DEVELOPMENT

The developer who connects the encryption function to the DBMS using the TOE shall ensure that the security functions of the TOE are securely applied in accordance with the requirements of guidance documents provided with the TOE.

OE.TIMESTAMP

The TOE shall receive a reliable timestamp from the operating system and record audit data relevant to operation of the TOE in an accurate fashion.

4 Extended Components Definition

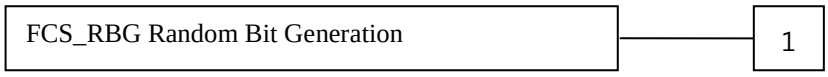
4.1 Cryptographic Support (FCS)

4.1.1 Random Bit Generation

Family Overview

This family (FCS_RBG, Random Bit Generation) defines requirements for generating random bits needed for TOE cryptographic operation.

Component Leveling and Description



FCS_RBG.1 Random Bit Generation requires the TSF to generate random bits needed for TOE cryptographic operation.

Management: FCS_RBG.1

There is no expected management requirement.

Audit: FCS_RBG.1

There is no expected audible action.

4.1.1.1 **FCS_RGB.1 Random Bit Generation**

Hierarchical to	No other components
Dependencies	No dependencies

FCS_RBG.1.1 The TSF shall generate random bits using the specified random bit generator that meets the following [assignment*: list of standards*].

4.2 Identification and Authentication (FIA)

4.2.1 TOE Internal Mutual Authentication

Family Overview

This family (FIA_IMA, TOE Internal Mutual Authentication) requires to provide the function of TOE internal mutual authentication during the user identification and authentication process.

Component Leveling and Description

FIA_IMA.1 TOE Internal Mutual Authentication requires to provide the function of TOE internal mutual authentication in the process of user identification and authentication.

Management: FIA_IMA.1

There is no expected management requirement.

Audit: FIA_IMA.1

The following actions are recommended to document as audit records if FAU_GEN Security Audit Data Generation family is included in the PP/ST.

a) Minimum : Success/failure of mutual authentication
b) Minimum : Modification of authentication protocol

4.2.1.1 FIA_IMA.1 TOE Internal Mutual Authentication

Hierarchical to	No other components
Dependencies	No dependencies

FIA_IMA.1 The TSF shall perform mutual authentication between [assignment: *parts of TOE*] using the [assignment: *authentication protocol*] that meets the following [assignment: *list of standards*].

4.3 User Data Protection (FDP)

4.3.1 User Data Encryption

Family Overview

This family (FDP_UDE User Data Encryption) provides the requirements to ensure confidentiality of user data.

Component Leveling and Description

FDP_UDE1 User Data Encryption requires to ensure confidentiality of user data.

Management: FDP_UDE.1

The following management function can be considered in the FMT.
a) Management of rules for encrypting/decrypting user data

Audit: FDP_UDE.1

The following action is recommended to document as audit records if FAU_GEN Security Audit Data Generation family is included in the PP/ST.

a) Minimum : Success and failure of encryption/decryption of user data

4.3.1.1 FDP_UDE.1 User Data Encryption

Hierarchical to	No other components
Dependencies	FCS_COP.1 Cryptographic Operation

FDP_UDE.1.1 The TSF shall provide users with the function of encrypting/decrypting user data in accordance with [assignment: *list of encryption/decryption methods*] specified.

4.4 Security Management (FMT)

4.4.1 ID and Password

Family Overview

This family (FMT_PWD, ID and Password) defines the requirements for the authorized administrator to control the management of IDs and passwords used in the TOE and to set up or modify an ID and/or password.

Component Leveling and Description

FMT_PWD.1 ID and Password Management requires the TSF to provide the function of managing IDs and passwords.

Management: FMT_PWD.1

The following management function can be considered in the FMT.
a) Management of rules for setting up an ID and password

Audit: FMT_PWD.1

The following action is recommended to document as audit records if FAU_GEN Security Audit Data Generation family is included in the PP/ST.

a) Minimum : All modifications to passwords

4.4.1.1 FMT_PWD.1 Management of ID and Password

Hierarchical to	No other components
Dependencies	FMT_SMF.1 Specification of Management Functions FMT_SMR.1 Security Roles
FMT_PWD.1.1	The TSF shall restrict its capability to manage passwords of [assignment: *list of functions*] to [assignment: *authorized roles*] as follows: 1. [assignment: *password combination rules and/or length*]; 2. [assignment: *other management such as management of special characters unusable for a password*].
FMT_PWD.1.2	The TSF shall restrict its capability to manage IDs of [assignment: *list of functions*] to [assignment: *authorized roles*] as follows*:* 1. [assignment: *ID combination rules and/or length*]; 2. [assignment : *other management such as management of special characters unusable for an ID*].
FMT_PWD.1.3	The TSF shall provide the function of [selection: *choose one of the following: setting up an ID and password during the installation process, setting up a password during installation, changing the ID and password during an authorized administrator's initial access, and changing the password during an authorized administrator's initial access*].

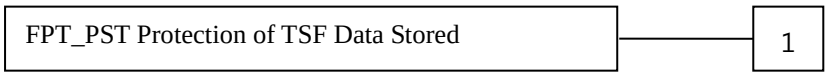
4.5 Production of the TSF (FPT)

4.5.1 Protection of TSF Data Stored

Family Overview

This family (FPT_PST, Protection of TSF Data Stored) defines rules to protect against unauthorized modification or disclosure of TSF data stored in repositories controlled by the TSF.

Component Leveling and Description



FPT_PST.1 Basic Protection of TSF Data Stored requires the protection of TSF data stored in repositories controlled by the TSF.

Management: FPT_PST.1
There is no expected management requirement.

Audit: FPT_PST.1
There is no expected audit requirement.

4.5.1.1 FPT_PST.1 Basic Protection of TSF Data Stored

Hierarchical to	No other components
Dependencies	No dependencies

FPT_PST.1.1 The TSF shall protect [assignment: *TSF data*] stored in repositories controlled by the TSF from unauthorized [selection: *disclosure*, *modification*].

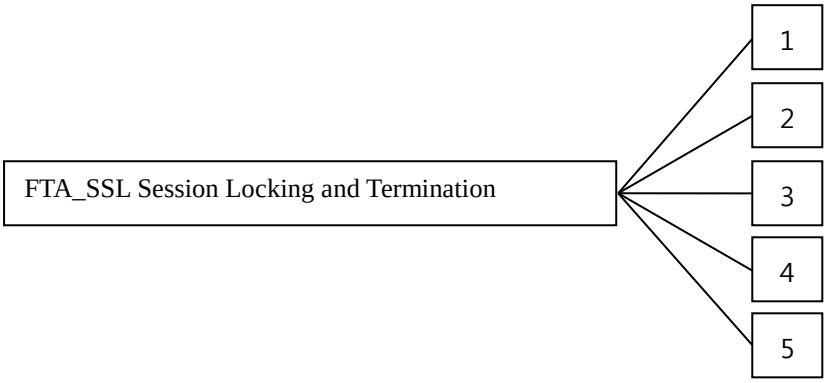
4.6 TOE Access (FTA)

4.6.1 Session Locking and Termination

Family Overview

This family (FTA_SSL, Session Locking and Termination) defines requirements for the TSF to provide its capacity of locking, unlocking and terminating TSF-initiated and user-initiated sessions.

Component Leveling and Description



In CC Part 2, Session Locking and Termination family consists of four components. In this PP, it is comprised of five components with one additional component as follows.

※ Omitted is the description regarding the four components included in CC Part 2.

FTA_SSL.5 TSF-initiated Session Management provides the requirements for the TSF to lock or terminate a session after a specified time of user inactivity.

Management: FTA_SSL.5

The following management function can be considered in the FMT.

a) Specification of user inactivity time that causes session locking or termination for each user
b) Specification of default user inactivity time that causes session locking or termination

Audit: FTA_SSL.5

The following action is recommended to document as audit records if FAU_GEN Security Audit Data Generation family is included in the PP/ST.

a) Minimum : Locking or termination of interactive sessions

4.6.1.1 FTA_SSL.5 TSF-initiated Session Management

Hierarchical to	No other components
Dependencies	[FIA_UAU.1 Authentication or no dependencies]

FTA_SSL.5.1 The TSF shall [selection :
• *lock a session and/or re-authenticate the user before unlocking the session, or*
• *terminate*] an interactive session after [assignment: user inactivity time].

5 Security Requirements

This Chapter describes security functional requirements and assurance requirements that shall be met in the TOE.

5.1 Security Functional Requirements

Security functional requirements defined in this ST are specified from relevant security functional components selected from CC Part 2, in order to satisfy the security objectives identified in Chapter 3. The following [Table 5-1] summarizes security functional components used in this ST.

[Table 5-1] Security Functional Requirements

Security Functional Class	Security Functional Component	
Security Audit (FAU)	FAU_ARP.1	Security Alarms
	FAU_GEN.1	Audit Data Generation
	FAU_SAA.1	Potential Violation Analysis
	FAU_SAR.1	Audit Review
	FAU_SAR.3	Selectable Audit Review
	FAU_STG.1	Protection of Audit Trail Storage
	FAU_STG.3	Action in Case of Possible Audit Data Loss
	FAU_STG.4	Prevention of Audit Data Loss
Cryptographic Support (FCS)	FCS_CKM.1(1)	Cryptographic Key Generation (User Data Encryption)
	FCS_CKM.1(2)	Cryptographic Key Generation (TSF Data Encryption)
	FCS_CKM.2	Cryptographic Key Distribution
	FCS_CKM.4	Cryptographic Key Destruction
	FCS_COP.1(1)	Cryptographic Operation (User Data Encryption)
	FCS_COP.1(2)	Cryptographic Operation (TSF Data Encryption)
	FCS_RBG.1	Random Bit Generation (extended)
User Data Protection (FDP)	FDP_UDE.1	User Data Encryption (extended)
	FDP_RIP.1	Protection of Partial Residual Information
Identification and Authentication (FIA)	FIA_AFL.1	Authentication Failure Handling
	FIA_IMA.1(1)	TOE Internal Mutual Authentication (btw. Management Server and Agent)
	FIA_IMA.1(2)	TOE Internal Mutual Authentication (btw. Management Server and Administrative Tool)
	FIA_SOS.1	Verification of Secrets
	FIA_UAU.1	Authentication
	FIA_UAU.4	Authentication Mechanism for Re-use Prevention
	FIA_UAU.7	Protection of Authentication Feedback
	FIA_UID.1	Identification
Security Management (FMT)	FMT_MOF.1	Management of Security Functions
	FMT_MTD.1	Management of TSF data
	FMT_PWD.1	Management of ID and Password
	FMT_SMF.1	Specification of Management Functions
	FMT_SMR.1	Security Roles
Protection of the TSF (FPT)	FPT_ITT.1	Basic Protection of Internally-transmitted TSF Data
	FPT_PST.	Basic Protection of TSF Data Stored (extended)
	FPT_TST.1	TSF Self-testing
TOE Access (FTA)	FTA_MCS.2	Limitation of Concurrent Session Number per User Attribute
	FTA_SSL.5	TSF-initiated Session Management (extended)
	FTA_TSE.1	TOE Session Establishment

5.1.1 Security Audit (FAU)

5.1.1.1 FAU_ARP.1 Security Alarms

Hierarchical to: No other components
Dependencies : FAU_SAA.1 Potential Violation Analysis

FAU_ARP.1.1 The TSF shall take action (see [Table 5-2] Actions against Security Violations] if any potential security violation is detected.

[Table 5-2] Actions against Security Violations

Security Functional Component	Security Violation	Action
FIA_AFL.1	- In case an administrator's authentication attempts fail consecutively for a defined number of times (default: 5 times)	- Inactivate the authentication function for a defined period (default: 5 mins.) - Send a warning message to the authorized administrator via email
FPT_TST.1	- In case the integrity verification fails - In case a self-test of the validated cryptographic module fails	- Send a warning message to the authorized administrator via email
FAU_STG.3	- In case audit trail exceeds the threshold (default: 90%)	- Send a warning massage to the authorized administrator via email
FAU_STG.4	- In case audit trail is full	- Overwrite obsolete and audited event data - Send a warning massage to the authorized administrator via email

5.1.1.2 FAU_GEN.1 Audit Data Generation

Hierarchical to : No other components
Dependencies : FPT_STM.1 Reliable Timestamp

FAU_GEN.1.1 The TSF shall be able to generate audit records of the following auditable events:
a) start-up and shut-down of audit functions;
b) all auditable events with *not specified* level of audit;
c) ["Auditable Event" in [Table 5-3], None].

FAU_GEN.1.2 The TSF shall document at least the following information within each audit record:
a) Date and time of an event, a type of an event, subject identity (if applicable) and an outcome (success or failure) of an event;
b) ["Additional Audit Record" in [Table 5-3], None] based on auditable event definition of functional components included in the PP/ST for each audit event type

[Table 5-3] Auditable Events

Security Functional Component	Auditable Event	Additional Audit Record
FAU_ARP.1	Action taken due to potential security violations	
FAU_SAA.1	Enabling and disabling of an analysis mechanism Automated responses performed by the tool	
FAU_STG.3	Action taken due to exceeding of a threshold	
FAU_STG.4	Action taken due to the audit storage failure	
FCS_CKM.1(1)	Success and failure of an action	
FCS_CKM.2	Success and failure of an action (only applicable to key distribution related to user data encryption/decryption)	
FCS_CKM.4	Success and failure of an action (only applicable to key destruction related to user data encryption/decryption)	
FCS_COP.1(1)	Success and failure of cryptographic operation, types of cryptographic operation	
FDP_UDE.1	Success and failure of user data encryption/decryption	policy name, encryption/decryption algorithm
FIA_AFL.1	Reaching to the threshold for the unsuccessful authentication attempts and action taken, If appropriate, subsequent, restoration to the normal state	
FIA_IMA.1	Success and failure of mutual authentication Modification of authentication protocol	DN of certificate subject
FIA_UAU.1	All use of authentication mechanism	

Security Functional Component	Auditable Event	Additional Audit Record
FIA_UAU.4	Re-use attempt of authentication data	
FIA_UID.1	All use of user identification mechanism including user identity provided	
FMT_MOF.1	All modifications to the TSF functions	
FMT_MTD.1	All modifications to TSF data values	Modified values of TSF data
FMT_PWD.1	All modifications to passwords	
FMT_SMF.1	Use of management functions	
FMT_SMR.1	Modifications to the user group of roles divided	
FPT_TST.1	Execution of the TSF self-tests and the results of the tests	Modified TSF data or executable codes in case of integrity violation
FTA_MCS.2	Denial of a new session based on the limitation of concurrent sessions	
FTA_SSL.5	Locking or termination of an interactive session	

5.1.1.3 FAU_SAA.1 Potential Violation Analysis

Hierarchical to : No other components
Dependencies : FAU_GEN.1 Audit Data Generation

FAU_SAA.1.1 The TSF shall be able to apply a set of rules while monitoring audited events; and, based on these rules, point out a potential violation during enforcement of the SFRs.

FAU_SAA.1.2 The TSF shall enforce the following rules for monitoring audited events:
a) accumulation or combination of [authentication failure audit event among auditable events in FIA_UAU.1, integrity violation audit event and failure event of the validated cryptographic module's self-test among auditable events in FPT_TST.1, [excess of threshold among auditable events in FAU_STG.3 and full audit trail event among auditable events in FAU_STG.4]] known as a potential security violation;
b) [None].

5.1.1.1 FAU_SAR.1 Audit Review

Hierarchical to : No other components
Dependencies : FAU_GEN.1 Audit Data Generation

FAU_SAR.1.1 The TSF shall provide [authorized administrator] with the capacity of reading [all the audit data] from audit records.

FAU_SAR.1.2 The TSF shall provide audit records in a way that is appropriate for the **authorized administrator** to interpret the information.

5.1.1.2 FAU_SAR.3 Selectable Audit Review

Hierarchical to : No other components
Dependencies : FAU_SAR.1 Audit Review

FAU_SAR.3.1 The TSF shall provide the capacity to apply [sorting in a descending order based on the time/date of events] of audit data based on [the "time/date of an event" AND "subject IP" AND "subject ID" and "an event type" AND "an event outcome"].

5.1.1.3 FAU_STG.1 Protection of Audit Trail Storage

Hierarchical to No other components
Dependencies FAU_GEN.1 Audit Data Generation
FAU_STG.1.1 The TSF shall protect audit records stored in audit trail from unauthorized deletion.
FAU_STG.1.2 The TSF shall *prevent* any unauthorized modification to audit records stored in audit trail.

5.1.1.4 FAU_STG.3 Action in Case of Possible Audit Data Loss

Hierarchical to : No other components

Dependencies : FAU_STG.1 Protection of Audit Trail Storage

FAU_STG.3.1 The TSF shall take action [notification to the authorized administrator, [none]] if the audit trail exceeds [limit defined as follows]:
a) Space used out of total audit trail storage (%), default: 90%
b) Criteria for permissible threshold entry : an integer between 10 to 90

5.1.1.5 **FAU_STG.4 Prevention of Audit Data Loss**

Hierarchical to : FAU_STG.3 Action in Case of Possible Audit Data Loss
Dependencies : FAU_STG.1 Protection of Audit Trail Storage

FAU_STG.4.1 The TSF shall *overwrite the most obsolete audit record* and [send an email to the authorized administrator] if audit trail is full.

5.1.2 Cryptographic Support (FCS)

5.1.2.1 **FCS_CKM.1(1) Cryptographic Key Generation (User Data Encryption)**

Hierarchical to : No other components
Dependencies: [FCS_CKM.2 Cryptographic Key Distribution or
FCS_COP.1 Cryptographic Operation]
FCS_CKM.4 Cryptographic Key Destruction

FCS_CKM.1.1 The TSF shall generate a cryptographic key in accordance with a specified cryptographic key generation algorithm [Cryptographic Algorithm in [Table 5-4]] and a specified cryptographic key length [Cryptographic Key Length in [Table 5-4]] that meet the following [[Table 5-4] List of Algorithm Standards and [Table 5-5] List of Random Bit Generation Standards]].

[Table 5-4] List of Algorithm Standards

Cls.	Cryptographic Algorithm	Cryptographic Key Length (bit)	Reference Standard
Block Cipher (Symmetric Key cryptography) User Date Encryption/Decryption	ARIA	128/192/256	KS X 1213-1
	SEED	128	TTAS.KO-12.0004/R1
	LEA	128/192/256	TTAK.KO-12.0223

[Table 5-5] List of Random Bit Generation Standards

Cls.	Random Bit Generation Algorithm	Reference Standard	Remark
Random Bit Generator	HASH_DRBG(SHA-256)	ISO/IEC 18031	Pseudorandom function

5.1.2.2 **FCS_CKM.1(2) Cryptographic Key Generation (TSF Data Encryption)**

Hierarchical to : No other components
Dependencies : [FCS_CKM.2 Cryptographic Key Distribution or
FCS_COP.1 Cryptographic Operation]
FCS_CKM.4 Cryptographic Key Destruction

FCS_CKM.1.1 The TSF shall generate a cryptographic key in a length specified in [Cryptographic Key Length in [Table 5-6]] that meets the following [[Table 5-6] List of Cryptographic Key Generation Standards, [Table 5-7] List of Random Bit Generation Standards].

[Table 5-6] List of Cryptographic Key Generation Standards

Cls.	Cryptographic Algorithm	Cryptographic Key Length (Bit)	Reference Standard
Block Cipher (Symmetrical Key Cryptography)	ARIA	256	KS X 1213-1

[Table 5-7] List of Random Bit Generation Standards

Cls.	Random Bit Generation Algorithm	Reference Standard	Remark
Random Bit Generator	HASH_DRBG(SHA-256)	ISO/IEC 18031	Pseudorandom function

5.1.2.3 **FCS_CKM.2 Cryptographic Key Distribution**

Hierarchical to : No other components
Dependencies : [FDP_ITC.1 Import of User Data without Security Attributes or
FDP_ITC.2 Import of User Data with Security Attributes or
FCS_CKM.1 Cryptographic Key Generation]
FCS_CKM.4 Cryptographic Key Destruction

FCS_CKM.2.1 The TSF shall distribute a cryptographic key in accordance with a cryptographic key distribution method specified in [Cryptographic Key Distribution Method in [Table 5-8]] that meets the following [[Table 5-8] List of Cryptographic Key Distribution Standards].

[Table 5-8] List of Cryptographic Key Distribution Standards

Usage	Cryptographic Algorithm	Cryptographic Key Length	Reference Standard
Public Key Cryptography	RSAES (SHA-256)	Public key 2048 bits	ISO/IEC 18033-2
Cryptographic Key Distribution Method			
- Distribution by encrypting an encryption key for information transmitted between the Management Server, Administrative Tool and Agent with the public key of the other server			

5.1.2.4 **FCS_CKM.4 Cryptographic Key Destruction**

Hierarchical to : No other components
Dependencies : [FDP_ITC.1 Import of User Data without Security Attributes or
FDP_ITC.2 Import of User Data with Security Attributes or
FCS_CKM.1 Cryptographic Key Generation]

FCS_CKM.4.1 The TSF shall destroy a cryptographic key in accordance with a specified cryptographic key destruction method [Deletion of a cryptographic key file and overwriting of memories with "0"] that meets the following [[Table 5-9] List of Cryptographic Key Destruction].

[Table 5-9] List of Cryptographic Key Destruction

Usage	Method	Cryptographic Key Length	Reference Standard
Cryptographic Key Destruction	Overwriting with 0 (0x00)	-	-
Function of Cryptographic Key Destruction			
- Destruction of a symmetric key for the use of KEKs (Key Encrypt Key) - Destruction of a symmetric key for the use of DEKs (Data Encrypt Key)			

5.1.2.5 **FCS_COP.1(1) Cryptographic Operation (User Data Encryption)**

Hierarchical to : No other components
Dependencies : [FDP_ITC.1 Import of User Data without Security Attributes or
FDP_ITC.2 Import of User Data with Security Attributes or
FCS_CKM.1 Cryptographic Key Generation]
FCS_CKM.4 Cryptographic Key Destruction

FCS_COP.1.1 The TSF shall perform [Cryptographic Operation List in [Table 5-10]] in accordance with a specified cryptographic algorithm [Cryptographic Algorithm in [Table 5-10]] and a specified cryptographic key length [Cryptographic Key Length in [Table 5-10]] that meets the following [[Table 5-10] List of Algorithm Standards].

[Table 5-10] List of Algorithm Standards

Cryptographic Operation List	Cryptographic Algorithm	Application Mode	Cryptographic Key Length (Bit)	Reference Standard
Block Cipher (Symmetric Key Cryptography) User data Encryption/ Decryption	ARIA	CBC	128/192/256	KS X 1213-1
	SEED	CBC	128	TTAS.KO-12.0004/R1
	LEA	CBC	128/192/256	TTAK.KO-12.0223
Hash Function (Uni-	SHA-256			ISO/IEC 10118-3

directional Cryptography) User Data Encryption	SHA-384			ISO/IEC 10118-3
	SHA-512			ISO/IEC 10118-3

5.1.2.6 **FCS_COP.1(2) Cryptographic Operation (TSF Data Encryption)**

FCS_COP.1(2) Cryptographic Operation (TSF Data Encryption)
Hierarchical to : No other components
Dependencies : [FDP_ITC.1 Import of User Data without Security Attributes or
FDP_ITC.2 Import of User Data with Security Attributes or
FCS_CKM.1 Cryptographic Key Generation]
FCS_CKM.4 Cryptographic Key Destruction

FCS_COP.1.1 The TSF shall perform [Cryptographic Operation List in [Table 5-11]] in accordance with a specified cryptographic key algorithm [Cryptographic Algorithm in [Table 5-11]] and a specified cryptographic key length [Cryptographic Key Length in [Table 5-11]] that meets the following [[Table 5-11] List of Cryptographic Operation Standards].

[Table 5-11] List of Cryptographic Operation Standards

Component	Cryptographic Operation List	Cryptographic Algorithm	Application Mode	Cryptographic Key Length(Bit)	Reference Standard
Management Server	Block Cipher (Symmetrical key cryptography) Encryption/decryption of user data cryptographic keys	ARIA	CBC	256	KS X 1213-1
	Block Cipher (Symmetrical key cryptography) Encryption/decryption of configuration information	ARIA	CBC	256	KS X 1213-1
	Hash function (Uni-directional cryptography) Encryption of administrator password	SHA-2		256	ISO/IEC 10118-3
	Block Cipher (Symmetrical key cryptography) Encryption/decryption of a session cryptographic key	ARIA	CBC	256	KS X 1213-1
	Block Cipher (Symmetrical key cryptography) Encryption/decryption of audit data	ARIA	CBC	256	KS X 1213-1
	Sign Mutual Authentication, TSF Data	RSA-PSS (SHA-256)		2048	PKCS #1 v2.1
	Sign verify Mutual Authentication, TSF Data	RSA-PSS (SHA-256)		2048	ISO/IEC 14888-2
	Integrity verification (TSF data, Encryption of TSF and User Data cryptographic keys)	HMAC-SHA256		256	ISO/IEC 9797-2
	TSF data derived key	PBKDF2(SHA-256)		256	TTAS.KO-12.0334
Agent	Block Cipher (Symmetrical key cryptography) Encryption/decryption of configuration information	ARIA	CBC	256	KS X 1213-1
	Block Cipher (Symmetrical key cryptography) Encryption/decryption of a session cryptographic key	ARIA	CBC	256	KS X 1213-1
	Block Cipher (Symmetrical key cryptography) Encryption/decryption of user data cryptographic keys	ARIA	CBC	256	KS X 1213-1
	Sign	RSA-PSS		2048	ISO/IEC

	Mutual Authentication, TSF Data	(SHA-256)			14888-2
	Sign verify Mutual Authentication, TSF Data	RSA-PSS (SHA-256)		2048	ISO/IEC 14888-2
	Integrity verification (TSF data, Encryption of TSF and User Data cryptographic keys)	HMAC- SHA256		256	ISO/IEC 9797-2
	TSF data derived key	PBKDF2(SH A-256)		256	TTAS.KO- 12.0334
Administrativ e Tool	Block Cipher (Symmetrical key cryptography) Encryption/decryption of configuration information	ARIA	CBC	256	KS X 1213-1
	Block Cipher (Symmetrical key cryptography) Encryption/decryption of a session cryptographic key	ARIA	CBC	256	KS X 1213-1
	Sign Mutual Authentication, TSF Data	RSA-PSS (SHA-256)		2048	ISO/IEC 14888-2
	Sign verify Mutual Authentication, TSF Data	RSA-PSS (SHA-256)		2048	ISO/IEC 14888-2
	Integrity verification (TSF data, Encryption of TSF Data cryptographic keys)	HMAC- SHA256		256	ISO/IEC 9797-2
	TSF data derived key	PBKDF2(SH A-256)		256	TTAS.KO- 12.0334

5.1.2.7 **FCS_RBG.1 Random Bit Generation (extended)**

Hierarchical to : No other components
Dependencies : No dependencies

FCS_RBG.1.1 The TSF shall generate a random bit using a specified random bit generator that meets the following [[Table 5-12] List of Random Bit Generation Standards].

[Table 5-12] List of Random Bit Generation Standards

Cls.	Random Bit Generation Algorithm	Reference Standard	Remark
Random Bit Generator	HASH_DRBG(SHA-256)	ISO/IEC 18031	Pseudorandom function

5.1.3 User Data Protection (FDP)

5.1.3.1 **FDP_UDE.1 User Data Encryption**

Hierarchical to : No other components
Dependencies : FCS_COP.1 Cryptographic Operation

FDP_UDE.1.1 The TSF shall provide TOE users with a function that enables encryption/decryption of user data in accordance with [Encryption/decryption method as per column, [none]] specified.

5.1.3.2 **FDP_RIP.1 Protection of Partial Residual Information**

Hierarchical to : No other components
Dependencies : No dependencies

FDP_RIP.1.1 The TSF shall ensure that all previous information of resources are not available in case the *resources are allocated and retrieved* to the following object [user data].

5.1.4 Identification and Authentication

5.1.4.1 **FIA_AFL.1 Authentication Failure Handling**

Hierarchical to : No other components
Dependencies : FIA_UAU.1 Authentication

FIA_AFL.1.1 In case there are *[1 ~ 5]* unsuccessful authentication attempts related to [administrator's authentication attempts], the TSF shall detect it.

FIA_AFL.1.2 When unsuccessful authentication attempts *have reached* to a defined number, the TSF shall [inactivate its identification and authentication function for five to ten minutes (default: fiveminutes)].

5.1.4.2 **FIA_IMA.1(1) TOE Internal Mutual Authentication**

Hierarchical to : No other components
Dependencies : No dependencies

FIA_IMA.1.1 The TSF shall perform mutual authentication through [self-implemented authentication protocol] that meets [None] between [the Management Server and Agent].

5.1.4.3 **FIA_IMA.1(2) TOE Internal Mutual Authentication**

Hierarchical to : No other components
Dependencies : No dependencies

FIA_IMA.1.1 The TSF shall perform mutual authentication through [self-implemented authentication protocol] that meets [None] between [the Management Server and Administrative Tool].

5.1.4.4 **FIA_SOS.1 Verification of Secrets**

Hierarchical to : No other components
Dependencies : No dependencies

FIA_SOS.1.1 The TSF shall provide a mechanism to verity that secrets meet the following [[Table 5-13] List of Secret Criteria].

[Table 5-13] List of Secret Criteria

Cls.	Defined Criteria
Valid Character	English characters (a-z, A~Z), numbers (0~9), special characters (~, `, !, @, #, $, %, ^, &, *, (,), -, _, +, =)
Combination Rule	Combination of all three character types
Valid Length	9 ~ 16 digits

5.1.4.5 **FIA_UAU.1 Authentication**

Hierarchical to : No other components
Dependencies : FIA_UID.1 Identification

FIA_UAU.1.1 The TSF shall allow [set-up of the Management Server IP and Port, exchanges of Nonce values and session key agreement] to be enforced on behalf of an **authorized administrator** before the **authorized administrator** is authenticated.

FIA_UAU.1.2 The TSF shall successfully authenticate an **authorized administrator** prior to the permission of all other actions mediated by the TSF on behalf of the **authorized administrator** besides actions specified in FIA_UAU.1.1.

5.1.4.6 **FIA_UAU.4 Authentication Mechanism of Re-use Prevention**

Hierarchical to : No other components
Dependencies : No dependencies

FIA_UAU.4.1 The TSF shall prevent re-use of authentication data related to [password authentication].

5.1.4.7 **FIA_UAU.7 Protection of Authentication Feedback**

Hierarchical to : No other components

Dependencies : FIA_UAU.1 Authentication

FIA_UAU.7.1 The TSF shall provide users with only the following [characters entered are masked with * when the administrator keys in secret information (password), feedback such as "the administrator has failed to log in" or "login failed for user" without the mention of a reason for an authentication failure] while the authentication is in progress.

5.1.4.8 **FIA_UID.1 Identification**

Hierarchical to : No other components
Dependencies : No dependencies

FIA_UID.1.1 The TSF shall allow [set-up of the Management Server IP and Port] to be enforced on behalf of an **authorized administrator** before the **authorized administrator** is identified.

FIA_UID.1.2 The TSF shall successfully identify an **authorized administrator** prior to the permission of all other actions mediated by the TSF on behalf of the **authorized administrator** besides actions specified in FIA_UID.1.1.

5.1.5 Security Management (FMT)

5.1.5.1 **FMT_MOF.1 Management of Security Functions**

Hierarchical to : No other components
Dependencies : FMT_SMF.1 Specification of Management Functions
FMT_SMR.1 Security Roles

FMT_MOF.1.1 The TSF shall restrict the capacity of performing ***management actions*** regarding the functions in [[Table 5-14] List of Authorized Administrator Functions] to the [authorized administrator].

[Table 5-14] List of Authorized Administrator Functions

Function	Management Action
Set-up and modification of administrator password	Determine and modify an action
Set-up and modification of audit information thresholds	Determine and modify an action
Modification of the number of authentication failures and inactivity time	Determine and modify an action
Set-up and modification of the mail server	Determine and modify an action
Set-up and modification of administrator access IP	Determine and modify an action
Real-time validation of modules	Initiate an action
View of audit information	Determine an action
Set-up and modification of cryptographic keys	Determine and modify an action
Establishment and modification of policies	Determine and modify an action
Column encryption	Determine an action

5.1.5.2 **FMT_MTD.1 TSF Data Management**

Hierarchical to : No other components
Dependencies : FMT_SMF.1 Specification of Management Functions
FMT_SMR.1 Security Roles

FMT_MTD.1.1 The TSF shall restrict the capacity of ***managing*** [Table 5-15] List of Authorized Administrator's TSF Data] to the [authorized administrator].

[Table 5-15] List of Authorized Administrator's TSF Data

TSF Data List	Management
Audit information	Query
Modification of administrator password	Modification
Set-up of audit thresholds	Query, modification
Set-up of the mail server	Query, modification
Administrator information	Query, modification

TSF Data List	Management
Set-up of administrator access IP	Query, modification
Cryptographic keys (user data encryption)	Query
Policy list	Query, modification
Authentication failure information	Query, modification
Column list	Query, modification

5.1.5.3 **FMT_PWD.1 Management of ID and Password**

Hierarchical to : No other components
Dependencies : FMT_SMF.1 Specification of Management Functions
FMT_SMR.1 Security Roles

FMT_PWD.1.1 The TSF shall restrict the capacity of managing [None] to [None].
1. [None]
2. [None]

FMT_PWD.1.2 The TSF shall restrict the capacity of managing ID of [None] to [None].
1. [None]
2. [None]

FMT_PWD.1.3 The TSF shall provide the function of *changing the password when the authorized administrator accesses for the first time.*

[Table 5-16] List of Secret Criteria

Cls.	Defined Criteria
Valid Characters	English characters (a-z, A~Z), numbers (0~9), special characters (~, `, !, @, #, $, %, ^, &, *, (,), -, _, +, =)
Combination Rule	Combination of all three character types
Valid Length	9 ~ 16 digits

5.1.5.4 **FMT_SMF.1 Specification of Management Functions**

Hierarchical to : No other components
Dependencies : No dependencies

FMT_SMF.1.1 The TSF shall be able to perform the following management functions: [list of management functions provided by the TSF]
[
a) List of security functions specified in FMT_MOF.1
b) List of TSF data management specified in FMT_MTD.1
c) List of functions specified in FMT_PWD.1
]

5.1.5.5 **FMT_SMR.1 Security Roles**

Hierarchical to : No other components
Dependencies : FIA_UID.1 Identification

FMT_SMR.1.1 The TSF shall maintain the [following roles of authorized users]:

[
Set-up and modification of administrator password
Set-up and modification of audit thresholds
Modification of the number of authentication failure and inactivity time
Set-up and modification of the mail server
Set-up and modification of administrator access IP
Real-time validation of modules

View of audit information
Set-up and modification of cryptographic keys
Establishment and modification of policies
Encryption of columns
]

FMT_SMR.1.2 The TSF shall be able to associate users with **the roles defined in FMT_SMR.1.1.**

5.1.6 Protection of the TSF (FPT)

5.1.6.1 **FPT_ITT.1 Basic Protection of Internally-transmitted TSF Data**

Hierarchical to : No other components
Dependencies : No dependencies

FPT_ITT.1.1 The TSF shall protect the TSF data from *disclosure and modification* **through encryption and verification of message integrity** when they are transmitted between discrete parts of the TOE.

5.1.6.2 **FPT_PST.1 Basic Protection of TSF Data Stored (extended)**

Hierarchical to : No other components
Dependencies : No dependencies

FPT_PST.1.1 The TSF shall protect [the following TSF data] stored in repositories controlled by the TSF from unauthorized *disclosure and modification.*

[
Administrator password
Set-up of audit thresholds
Set-up of the mail server
Administrator information
Set-up of administrator access IP
Cryptographic keys (User data encryption)
Policy list
Authentication failure information
Column list
]

5.1.6.3 **FPT_TST.1 TSF Self-testing**

Hierarchical to : No other components
Dependencies : No dependencies

FPT_TST.1.1 The TSF shall run a self-test *during the initial start-up and on a periodical basis during normal operation* to demonstrate accurate operation of *[[Table 5-17] Items of Self-Test*].

FPT_TST.1.2 The TSF shall provide the **authorized administrator** with the capacity of verifying integrity of [*configuration file in [Table 5-18] Items of TOE Integrity Test].*

FPT_TST.1.3 The TSF shall provide the **authorized administrator** with the capacity of verifying integrity of [*execution file in [Table 5-18] Items of TOE Integrity Test].*

[Table 5-17] Items of Self-Test

TOE Classification	Item of Self-test Item	Testing
Management Server	Validated Cryptographic Module MagicCrypto V2.2.0	Self-test internally performed by the validated cryptographic module
	Process Name MagicDBPolicy	Check if main processes needed to run the Management Server are in normal operation and send the result to audit logs
Agent	Validated Cryptographic Module	Self-test internally performed by the validated cryptographic module

TOE Classification	Item of Self-test Item	Testing
	MagicCrypto V2.2.0	
	Process Name MDBAgent	Check if main processes needed to run the Agent are in normal operation and send the result to audit logs
Administrative Tool	Validated Cryptographic Module MagicCrypto V2.2.0	Self-test internally performed by the validated cryptographic module
	Process Name MagicDBPlus_v2.0_Admin	Check if main processes needed to run the Administrative Tool are in normal operation and send the result to audit logs

[Table 5-18] Items of TOE Integrity Test

TOE Classification	Integrity Test Item	Testing
Management Server	MagicDBPolicy.conf	The configuration information is verified with a signature value using the certificate issued during installation of the Management Server, and HMAC generated with a KEK is stored when the configuration file is created. An integrity test is performed to see if there is any falsification/counterfeit due to an unauthorized user during the start-up of the Management Server, upon the request of the administrator using the Administrative Tool and every three hours since the start-up of the Management Server. After the integrity test, audit logs are sent via email.
	libMagicCrypto.so start.sh stop.sh restart.sh	Hash (SHA256) values from libMagicCrypto.so, start.sh, stop.sh, restart.sh installed during generation of the configuration file are stored in the configuration file. An integrity test is performed to see if there is any falsification due to an unauthorized user during the start-up of the Management Server, upon the request of the administrator using the Administrative Tool and every three hours since the start-up of the Management Server. After the integrity test, audit logs are sent via email.
	magicdb.dat magicdb.audit.dat	HMAC generated with a KEK for each block in a certain size is stored during recording of audit log data. An integrity test is performed to see if there is any falsification due to an unauthorized user during the start-up of the Management Server, upon the request of the administrator using the Administrative Tool and every three hours since the start-up of the Management Server. After the integrity test, audit logs are sent via email.
	Cryptographic File	HMAC generated as a KEK is stored in the file when cryptographic keys are generated. An integrity test is performed to see if there is any falsification due to an unauthorized user during the start-up of the Management Server, upon the request of the administrator using the Administrative Tool and every three hours since the start-up of the Management Server. After the integrity test, audit logs are sent via email.
	Log File	If a log is added to a log file, a chain verification is needed using a hash (SHA256) for each line. An integrity test is performed to see if there is any falsification due to an unauthorized user upon the request of the administrator using the Administrative Tool and every three hours since the start-up of the Management Server. After the integrity test, audit logs are sent via email.
Agent	MDBAgent.conf	The configuration information is verified with a signature value using the certificate issued during installation of the Agent, and HMAC generated with a KEK is stored when the configuration file is created. An integrity test is performed to see if there is any falsification/counterfeit due to an unauthorized user during the start-up of the Management Server, upon the request of the administrator using the Administrative Tool and every three

TOE Classification	Integrity Test Item	Testing
		hours since the start-up of the Management Server. After the integrity test, audit logs are sent via email.
	libMagicDB.so libMagicCrypto.so start.sh stop.sh restart.sh	Hash (SHA256) values from libMagicDB.so, libMagicCrypto.so, start.sh, stop.sh, restart.sh installed during generation of the configuration file are stored in the configuration file. An integrity test is performed to see if there is any falsification due to an unauthorized user during the start-up of the Management Server, upon the request of the administrator using the Administrative Tool and every three hours since the start-up of the Management Server. After the integrity test, audit logs are sent via email.
	Log File	HMAC generated with a KEK for each block in a certain size is stored during recording of audit log data. An integrity test is performed to see if there is any falsification due to an unauthorized user during the start-up of the Management Server, upon the request of the administrator using the Administrative Tool and every three hours since the start-up of the Management Server. After the integrity test, audit logs are sent via email.
	DEK	A DEK for user data delivered to the Management Server is encrypted (ARIA-256-CBC) and stored in the memories and, while being stored, its integrity is ensured using HMAC values generated with a KEK. An integrity test is performed to see if there is any falsification due to an unauthorized user during the start-up of the Management Server, upon the request of the administrator using the Administrative Tool and every three hours since the start-up of the Management Server. After the integrity test, audit logs are sent via email.
Administrative Tool	MDBAdmin.xml.enc	The configuration information is verified with a signature value using the certificate issued during installation of the Administrative Tool, and HMAC generated with a KEK is stored when the configuration file is created. An integrity test is performed to see if there is any falsification/counterfeit due to an unauthorized user during the start-up of the Management Server, upon the request of the administrator using the Administrative Tool and every three hours since the start-up of the Management Server. After the integrity test, audit logs are sent via email.
	MagicCryptoV22.dll install.ico uninstall.ico Uninstall.exe	Hash (SHA256) values from MagicCryptoV22.dll, install.ico, uninstall.ico, Uninstall.exe installed during generation of the configuration file are stored in the configuration file. An integrity test is performed to see if there is any falsification due to an unauthorized user during the start-up of the Management Server, upon the request of the administrator using the Administrative Tool and every three hours since the start-up of the Management Server. After the integrity test, audit logs are sent via email.

5.1.7 TOE Access (FTA)

5.1.7.1 **FTA_MCS.2 Limitation of Concurrent Session Number per User Attribute**

Hierarchical to : FTA_MCS.1 Basic Limitation of Concurrent Session Number
Dependencies : FIA_UID.1 Identification

FTA_MCS.2.1 The TSF shall restrict the maximum number of concurrent sessions that belong to the same **administrator** in accordance with the rules [with respect to the list of management functions defined in FMT_SMF.1.1:

a) the maximum number of concurrent sessions shall be restricted to 1 on the management access by the same administrator with authority over “Management Actions” in FMT_MOF.1.1 and “Management” in FMT_MTD.1.1.

b) the maximum number of concurrent sessions shall be restricted to {0} on the management access by the same administrator with authority to perform only query out of "Management" in FMT_MTD.1.1 with no authority over "Management Actions" in FMT_MOF.1.1
c) [None]
].

FTA_MCS.2.2 The TSF shall enforce, by default, the limitation of a session number per **administrator** to [1].

5.1.7.2 **FTA_SSL.5 TSF-initiated Session Management (extended)**

Hierarchical to : No other components
Dependencies : FIA_UAU.1 Authentication or no dependencies

FTA_SSL.5.1 The TSF shall *terminate an interactive session* after [**administrator**'s inactivity time: 10 minutes].

5.1.7.3 **FTA_TSE.1 TOE Session Establishment**

Hierarchical to : No other components
Dependencies : No dependencies

FTA_TSE.1.1 The TSF shall be able to deny establishment of the administrator's management access session based on [access IP, [None]].

5.2 Security Assurance Requirements

Security assurance requirements in this ST are comprised of assurance components in CC Part 3 and the evaluation assurance level is EAL1+. Assurance components are summarized in [Table 5-19].

[Table 5-19] Assurance Requirements

Assurance Class	Assurance Component	
ST Evaluation	ASE_INT.1	ST Introduction
	ASE_CCL.1	Conformance Claim
	ASE_OBJ.1	Security Objectives for Operational Environment
	ASE_ECD.1	Extended Components Definition
	ASE_REQ.1	Stated Security Requirements
	ASE_TSS.1	TOE Summary Specification
Development	ADV_FSP.1	Basic Functional Specification
Guidance Documents	AGD_OPE.1	Operational User Guidance
	AGD_PRE.1	Preparation Procedures
Life-cycle Support	ALC_CMC.1	TOE Leveling
	ALC_CMS.1	TOE Configuration Management (CM) Coverage
Tests	ATE_FUN.1	Functional Testing
	ATE_IND.1	Independent testing : conformance
Vulnerability Assessment	AVA_VAN.1	Vulnerability Survey

5.2.1 Security Target Evaluation

5.2.1.1 **ASE_INT.1 ST Introduction**

Dependencies : No dependencies

Developer Action Elements

ASE_INT.1.1D The developer shall provide a ST introduction.

Content and Presentation Elements

ASE_INT.1.1C The ST introduction shall contain a ST reference, a TOE reference, a TOE overview and a TOE description.

ASE_INT.1.2C The ST reference shall uniquely identify the ST.

ASE_INT.1.3C The TOE reference shall uniquely identify the TOE.

ASE_INT.1.4C The TOE overview shall summarize the usage and major security features of the TOE.

ASE_INT.1.5C The TOE overview shall identify the TOE type.

ASE_INT.1.6C The TOE overview shall identify any non-TOE hardware/software/firmware required by the TOE.

ASE_INT.1.7C The TOE description shall describe the physical scope of the TOE.

ASE_INT.1.8C The TOE description shall describe the logical scope of the TOE.

Evaluator Action Elements

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

ASE_INT.1.2E The evaluator shall confirm that the TOE reference, the TOE overview and the TOE description are consistent with each other**.**

5.2.1.2 **ASE_CCL.1 Conformance Claim**

Dependencies : ASE_INT.1 ST Introduction

ASE_ECD.1 Extended Components Definition
ASE_REQ.1 Stated Security Requirements

Developer Action Elements

ASE_CCL.1.1D The developer shall provide a conformance claim.

ASE_CCL.1.2D The developer shall provide a conformance claim rationale.

Content and Presentation Elements

ASE_CCL.1.1C The conformance claim shall contain a CC conformance claim to identify the version of the CC to which the ST and the TOE conform.

ASE_CCL.1.2C The CC conformance claim shall describe the conformance of the ST to CC Part 2 as either "CC Part 2 conformant" or "CC Part 2 extended."

ASE_CCL.1.3C The CC conformance claim shall describe the conformance of the ST to CC Part 3 as either "CC Part 3 conformant" or "CC Part 3 extended."

ASE_CCL.1.4C The CC conformance claim shall be consistent with the extended components definition.

ASE_CCL.1.5C The conformance claim shall identify all PPs and security requirements packages to which the ST conforms.

ASE_CCL.1.6C The conformance claim shall describe any conformance of the ST to a package as either "package-conformance" or "package-augmented."

ASE_CCL.1.7C The conformance claim rationale shall demonstrate that the TOE type of the ST is consistent with the TOE type in the PPs to which the ST conforms.

ASE_CCL.1.8C The conformance claim rationale shall demonstrate that the statement of the security problem definition in the ST is consistent with the statement of the security problem definition in the PPs to which the ST conforms.

ASE_CCL.1.9C The conformance claim rationale shall demonstrate that the statement of security objectives in the ST is consistent with the statement of security objectives in the PPs to which the ST conforms.

ASE_CCL.1.10C The conformance claim rationale shall demonstrate that the statement of security requirements in the ST is consistent with the statement of security requirement in the PPs to which the ST conforms.

Evaluator Action Elements

ASE_CCL.1.1E The evaluator shall confirm that the information provided meets all requirements for contents and presentation of evidence.

5.2.1.3 **ASE_OBJ.1 Security Objectives for Operational Environment**

Dependencies : No dependencies

Developer Action Elements

ASE_OBJ.1D The developer shall provide a statement of security objectives.

Content and Presentation Elements

ASE_OBJ.1C The statement of security objectives shall describe the security objectives for the operational environment.

Evaluator Action Elements

ASE_OBJ.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.1.4 **ASE_ECD.1 Extended Components Definition**

Dependencies : No dependencies

Developer Action Elements

ASE_ECD.1.1D The developer shall provide a statement of security requirements.

ASE_ECD.1.2D The developer shall provide an extended components definition.

Content and Presentation Elements

ASE_ECD.1.1C The statement of security requirements shall identify all extended security requirements.

ASE_ECD.1.2C The extended components definition shall define an extended component for each extended security requirement.

ASE_ECD.1.3C The extended components definition shall describe how each extended component is related to the existing CC components, families and classes.

ASE_ECD.1.4C The extended components definition shall use the existing CC components, families and methodologies as a model for presentation.

ASE_ECD.1.5C The extended components shall consist of measurable and objective elements such that conformance or non-conformance to these elements can be demonstrated.

Evaluator Action Elements

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

ASE_ECD.1.2E The evaluator shall confirm that no extended component can be clearly expressed using existing components.

5.2.1.5 **ASE_REQ.1 Stated Security Requirements**

Dependencies : ASE_ECD.1 Extended Components Definition

Developer Action Elements

ASE_REQ.1.1D The developer shall provide a statement of security requirements.

ASE_REQ.1.2D The developer shall provide a security requirement rationale.

Content and Presentation Elements

ASE_REQ.1.1C The statement of security requirements shall describe the SFRs and the SARs.

ASE_REQ.1.2C .All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.

ASE_REQ.1.3C The statement of security requirements shall identify all operations on the security requirements.

ASE_REQ.1.4C All operations shall be performed correctly.

ASE_REQ.1.5C Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.

ASE_REQ.1.6C The statement of security requirements shall be internally consistent.

Evaluator Action Elements

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

5.2.1.6 **ASE_TSS.1 TOE Summary Specification**

Dependencies : ASE_INT.1 ST Introduction
ASE_REQ.1 Stated Security Requirements
ADV_FSP.1 Basic Functional Specification

Developer Action Elements

ASE_TSS.1.1D **The developer shall provide a TOE summary specification.**

Content and Presentation Elements

ASE_TSS.1.1C The TOE summary specification shall describe how the TOE meets each SFR.

Evaluator Action Elements

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

ASE_TSS.1.2E The evaluator shall confirm that the TOE summary specification is consistent with the TOE overview and the TOE description.

5.2.2 Development

5.2.2.1 **ADV_FSP.1 Security-enforcing Functional Specification**

Dependencies : No dependencies

Developer Action Elements

ADV_FSP.1.1D The developer shall provide a functional specification.

ADV_FSP.1.2D The developer shall provide a tracing from the functional specification to the SFRs.

Content and Presentation Elements

ADV_FSP.1.1C The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.2C The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.3C The functional specification shall provide rationale for the implicit categorization of interfaces as SFR-non-interfering.

ADV_FSP.1.4C The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

Evaluator Action Elements

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 SFRs.

5.2.3 Guidance Documents

5.2.3.1 **AGD_OPE.1 Operational User Guidance**

Dependencies : ADV_FSP.1 Basic Functional Specification

Developer Action Elements

AGD_OPE.1.1D The developer shall provide operational user guidance.

Content and Presentation Elements

AGD_OPE.1.1C The operational user guidance shall describe, for each user role, the user-accessible

functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2C The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3C The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4C The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including a change in the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5C The operational user guidance 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.

AGD_OPE.1.6C The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfil the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7C The operational user guidance shall be clear and reasonable.

Evaluator Action Elements

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

5.2.3.2 **AGD_PRE.1 Preparative Procedures**

Dependencies : No dependencies

Developer Action Elements

AGD_PRE.1.1D The developer shall provide the TOE including its preparative procedures.

Content and Presentation Elements

AGD_PRE.1.1C The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE.1.2C The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

Evaluator Action Elements

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

AGD_PRE.1.2E The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

5.2.4 Life-cycle Support

5.2.4.1 **ALC_CMC.1 Labelling of the TOE**

Dependencies : ALC_CMS.1 TOE CM Coverage

Developer Action Elements

ALC_CMC.1.1D The developer shall provide the TOE and a reference to the TOE.

Content and Presentation Elements

ALC_CMC.1.1C The TOE shall be labelled with its unique reference.

Evaluator Action Elements

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

5.2.4.2 **ALC_CMS.1 TOE CM Coverage**

Dependencies: No dependencies

Developer Action Elements

ALC_CMS.1.1D The developer shall provide a configuration list for the TOE.

Content and Presentation Elements

ALC_CMS.1.1C The configuration list shall include the following: the TOE itself and the evaluation evidence required by the SARs.

ALC_CMS.1.2C The configuration list shall uniquely identify the configuration items.

Evaluator Action Elements

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

5.2.5 Tests

5.2.5.1 **ATE_FUN.1 Functional Testing**

Dependencies : ATE_COV.1 Evidence of Test Coverage

Developer Action Elements

ATE_FUN.1.1D The developer shall test the TSF and document the results.

ATE_FUN.1.2D The developer shall provide test documentation.

Content and Presentation Elements

ATE_FUN.1.1C The test documentation shall consist of test plans, expected test results and actual test results.

ATE_FUN.1.2C The test plans shall identify the tests to be performed and described the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.3C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.4C The actual test results shall be consistent with the expected test results.

Evaluator Action Elements

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

5.2.5.2 **ATE_IND.1 Independent Testing : Conformance**

Dependencies : ADV_FSP.1 Basic Functional Specification
AGD_OPE.1 Operational User Guidance
AGD_PRE.1 Preparative Procedures

Developer Action Elements

ATE_IND.1.1D The developer shall provide the TOE to be tested.

Content and Presentation Elements

ATE_IND.1.1C The TOE shall be suitable for testing.

Evaluator Action Elements

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

ATE_IND.1.2E The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

5.2.6 Vulnerability Assessment

5.2.6.1 **AVA_VAN.1 Vulnerability Survey**

Dependencies : ADV_FSP.1 Basic Functional Specification
AGD_OPE.1 Operational User Guidance
AGD_PRE.1 Preparative Procedures

Developer Action Elements

AVA_VAN.1.1D The developer shall provide the TOE to be tested.

Content and Presentation Elements

AVA_VAN.1.1C The TOE shall be suitable for testing.

Evaluator Action Elements

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

AVA_VAN.1.2E The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.1.3E The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker with basic attack potentials.

5.3 Security Requirement Rationale

The security requirement rationale demonstrates that the SFRs described are suitable to satisfy the security objectives and, consequently, are appropriate to address the security problems.

5.3.1 Dependency of the SFRs

Rationale of the following functional components' dependency is shown in [Table 5-20] below.

[Table 5-20] Dependency Rationale

No.	Functional Component	Dependency	Reference No.
1	FAU_ARP.1	FAU_SAA.1	3
2	FAU_GEN.1	FPT_STM.1	OE. TIMESTAMP
3	FAU_SAA.1	FAU_GEN.1	2
4	FAU_SAR.1	FAU_GEN.1	2
5	FAU_SAR.3	FAU_SAR.1	4
6	FAU_STG.1	FAU_GEN.1	2
7	FAU_STG.3	FAU_STG.1	6
8	FAU_STG.4	FAU_STG.1	6
9	FCS_CKM.1(1)	FCS_CKM.2 or FCS_COP.1 FCS_CKM.4	11, 13 12
10	FCS_CKM.1(2)	FCS_CKM.2 or FCS_COP.1 FCS_CKM.4	11, 14 12
11	FCS_CKM.2	FDP_ITC.1, FDP_ITC.2 or FCS_CKM.1 FCS_CKM.4	9, 10 12
12	FCS_CKM.4	FDP_ITC.1, FDP_ITC.2 or FCS_CKM.1	9, 10
13	FCS_COP.1(1)	FDP_ITC.1, FDP_ITC.2 or FCS_CKM.1 FCS_CKM.4	9 12
14	FCS_COP.1(2)	FDP_ITC.1, FDP_ITC.2 or FCS_CKM.1 FCS_CKM.4	10 12
15	FCS_RBG.1	-	-
16	FDP_UDE.1	FCS_COP.1	13
17	FDP_RIP.1	-	-
18	FIA_AFL.1	FIA_UAU.1	22
19	FIA_IMA.1(1)	-	-
20	FIA_IMA.1(2)	-	-
21	FIA_SOS.1	-	-
22	FIA_UAU.1	FIA_UID.1	25
23	FIA_UAU.4	-	-
24	FIA_UAU.7	FIA_UAU.1	22
25	FIA_UID.1	-	-
26	FMT_MOF.1	FMT_SMF.1 FMT_SMR.1	29 30
27	FMT_MTD.1	FMT_SMF.1 FMT_SMR.1	29 30
28	FMT_PWD.1	FMT_SMF.1 FMT_SMR.1	29 30
29	FMT_SMF.1	-	-
30	FMT_SMR.1	FIA_UID.1	25
31	FPT_ITT.1	-	-
32	FPT_PST.1	-	-
33	FPT_TST.1	-	-
34	FTA_MCS.2	FIA_UID.1	25
35	FTA_SSL.5	FIA_UAU.1	22
36	FTA_TSE.1	-	-

FAU_GEN.1 has a dependency on FPT_STM.1. However, the TOE uses reliable timestamps provided in the TOE operational environment and accurately records audit data related to the operation of the TOE. Thus, the

dependency of FAU_GEN.1 is satisfied by OE. TIMESTAMP which is a security objective for the operation environment in lieu of FPT_STM.1.

FAU_STG.3 and FAU_STG.4 have a dependency on FAU_STG.1. The TOE stores audit data related to the TOE operation and ensures that audit data are protected from unauthorized deletion or modification, thereby satisfying the dependency of FAU_STG.1.

5.3.2 Dependency Rationale of Security Assurance Requirements

As the dependency of EAL1 Assurance Package provided in the CC is already satisfied, its rationale is omitted herein.

The augmented assurance requirement, ATE_FUN.1, has a dependency on ATE_COV.1. ATE_FUN.1 has been augmented to ensure that the developer performs a test on test items and documents it in the test transcript in an accurate manner. However, ATE_COV.1 is not included in this ST since the PP considers that it is not necessarily required to include ATE_COV.1 that presents the consistency between test items and TSFI.

6 TOE Summary Specification

This Chapter describes security functionality required by the TOE and the list of security functionality is shown in [Table 6-1].

[Table 6-1] List of Security Functions

Security Functional Class	Security Functional Component	
Security Audit (FAU)	FAU_ARP.1	Security Alarms
	FAU_GEN.1	Audit Data Generation
	FAU_SAA.1	Potential Violation Analysis
	FAU_SAR.1	Audit Review
	FAU_SAR.3	Selectable Audit Review
	FAU_STG.1	Protection of Audit Trail Storage
	FAU_STG.3	Action in Case of Possible Audit Data Loss
	FAU_STG.4	Prevention of Audit Data Loss
Cryptographic Support (FCS)	FCS_CKM.1(1)	Cryptographic Key generation (User Data Encryption)
	FCS_CKM.1(2)	Cryptographic Key Generation (TSF Data Encryption)
	FCS_CKM.2	Cryptographic Key Distribution
	FCS_CKM.4	Cryptographic Key Destruction
	FCS_COP.1(1)	Cryptographic Operation (User Data Encryption)
	FCS_COP.1(2)	Cryptographic Operation (TSF Data Encryption)
	FCS_RGB.1(extended)	Random Bit Generation
	FDP_UDE.1	User Data Encryption
	FDP_RIP.1	Protection of Partial Residual Information
Identification and Authentication (FIA)	FIA_AFL.1	Authentication Failure Handling
	FIA_IMA.1(1)	TOE Internal Mutual Authentication
	FIA_IMA.1(2)	TOE Internal Mutual Authentication
	FIA_SOS.1	Verification of Secrets
	FIA_UAU.1	Authentication
	FIA_UAU.4	Authentication Mechanism for Re-use Prevention
	FIA_UAU.7	Protection of Authentication Feedback
	FIA_UID.1	Identification
Security Management (FMT)	FMT_MOF.1	Management of Security Functions
	FMT_MTD.1	TSF Data Management
	FMT_PWD.1	Management of ID and Password
	FMT_SMF.1	Specification of Management Functions
	FMT_SMR.1	Security Roles
Protection of the TSF (FPT)	FPT_ITT.1	Basic Protection of Internally-transmitted TSF Data
	FPT_PST.1(extended)	Basic Protection of TSF Data Stored
	FPT_TST.1	TSF Self-testing
TOE Access (FTA)	FTA_MCS.2	Limitation of Concurrent Session Number per User Attribute
	FTA_SSL.5(extended)	TSF-initiated Session Management
	FTA_TSE.1	TOE Session Establishment

6.1 Security Audit

The TOE records and manages all audit data that can occur during operation in the Management Server such as start/end of an audit function, success/failure of identification and authentication of an administrator, details of the TOE configuration changes and security function execution.

The audit data information is defined in [Table 6-2] Auditable Events.

[Table 6-2] Auditable Events

Security Functional Component	Auditable Event	Additional Audit Record

Security Functional Component	Auditable Event	Additional Audit Record
FAU_ARP.1	Action taken due to potential security violations	
FAU_SAA.1	Enabling and disabling of an analysis mechanism Automated responses performed by the tool	
FAU_STG.1	Protection of audit trail storage	
FAU_STG.3	Action taken due to exceeding of a threshold	
FAU_STG.4	Action taken due to the audit storage failure	
FCS_CKM.1(1)	Success and failure of an action	
FCS_CKM.2	Success and failure of an action (only applicable to key distribution related to user data encryption/decryption)	
FCS_CKM.4	Success and failure of an action (only applicable to key destruction related to user data encryption/decryption)	
FCS_COP.1(1)	Success and failure of cryptographic operation, types of cryptographic operation	
FDP_UDE.1	Success and failure of user data encryption/decryption	policy name, encryption/decryption algorithm
FIA_AFL.1	Reaching to the threshold for the unsuccessful authentication attempts and action taken, If appropriate, subsequent restoration to the normal state	
FIA_IMA.1	Success and failure of mutual authentication Modification of authentication protocol	DN of certificate subject
FIA_UAU.1	All use of authentication mechanism	
FIA_UAU.4	Re-use attempt of authentication data	
FIA_UID.1	All use of user identification mechanism including user identity provided	
FMT_MOF.1	All modifications to the TSF functions	
FMT_MTD.1	All modifications to TSF data values	Modified values of TSF data
FMT_PWD.1	All modifications to passwords	
FMT_SMF.1	Use of management functions	
FMT_SMR.1	Modifications to the user group of roles divided	
FPT_TST.1	Execution of the TSF self-tests and the results of the tests	Modified TSF data or executable codes in case of integrity violation
FTA_MCS.2	Denial of a new session based on the limitation of concurrent sessions	
FTA_SSL.5	Locking or termination of an interactive session	

The TOE blocks access from unauthorized administrator for protection of audit trail storage and safeguards audit data stored.

If the storage where the audit data are stored exceeds the threshold defined by the administrator (defined as an integer between 10 and 90, default value: 90, unit: %), a warning message is sent to the authorized administrator via email.

If the audit data storage is full, the obsolete and audited event data are overwritten and a warning message is sent to the authorized administrator via email.

The authorized administrator can review audit data through the Administrative Tool (GUI) and retrieve audit data in accordance with date/time of an event, an event type, and an outcome of an event. Search results of the audit data can be sorted and displayed in a descending order based on the date/time of events. The function of modifying/deleting audit data is not provided.

If the TOE detects any potential security violation, an action against the security violation is conducted as shown in [Table 6-3].

[Table 6-3] Actions against Security Violations

Security Functional Component	Security Violation	Action
FIA_AFL.1	- In case an administrator's authentication attempts fail consecutively for a defined number of times (default: 5 times)	- Inactivate the authentication function for a defined period (default: 5 mins.) - Send a warning message to the authorized administrator via email
FPT_TST.1	- In case the integrity verification fails - In case a self-test of the validated cryptographic module fails	- Send a warning message to the authorized administrator via email
FAU_STG.3	- In case audit trail exceeds the threshold (default: 90%)	- Send a warning message to the authorized administrator via email
FAU_STG.4	- In case audit trail is full	- Overwrite obsolete and audited event data - Send a warning message to the authorized administrator via email

Relevant SFR : FAU_STG.1, FAU_ARP.1, FAU_GEN.1, FAU_SAA.1, FAU_SAR.1, FAU_SAR.3, FAU_STG.3, FAU_STG.4

6.2 Cryptographic Support

The TOE uses MagicCrypto V2.2.0 as a validated cryptographic module as shown in [Table 6-4] Validated Cryptographic Module.

The TOE generates a random bit using a random bit generation algorithm in [Table 6-6] while performing "cryptographic key generation of user data" and generates a cryptographic key from user data in accordance with a symmetrical key cryptography algorithm in [Table 6-5] and a cryptographic key length in [Table 6-5].

[Table 6-4] Validated Cryptographic Module

Name of Cryptographic Module	Validation No.	Developer	Date Validated
MagicCrypto V2.2.0	CM-162-2025.3	Dream Security Co., Ltd.	2020-03-03

[Table 6-5] List of Algorithm Standards

Cls.	Symmetrical Key Cryptography Algorithm	Cryptographic Key Length (bit)	Reference Standard
Block cipher (symmetrical key cryptography) User data encryption/decryption	ARIA	128/192/256	KS X 1213-1
	SEED	128	TTAS.KO-12.0004/R1
	LEA	128/192/256	TTAK.KO-12.0223

[Table 6-6] List of Random Bit Generation Standards

Cls.	Random Bit Generation Algorithm	Reference Standard	Remark
Random Bit Generator	HASH_DRBG(SHA-256)	ISO/IEC 18031	Pseudorandom function

The TOE generates a random bit using a random bit generation algorithm in [Table 6-4] while performing "cryptographic key generation of TSF data" and generates a TSF cryptographic key from TSF data in accordance with ARIA algorithm and a cryptographic bit length (256 bits) under a block cipher method (symmetrical key cryptography).

The TOE performs a cryptographic operation specified in the cryptographic operation list of [Table 6-7] in accordance with a cryptographic algorithm and a cryptographic key length specified in [Table 6-7] while performing "cryptographic operation of user data."

[Table 6-7] List of Algorithm Standards

Cryptographic Operation List	Cryptographic Algorithm	Application Mode	Cryptographic Key Length (Bit)	Reference Standard
Block Cipher (Symmetric Key Cryptography) User data Encryption/ Decryption	ARIA	CBC	128/192/256	KS X 1213-1
	SEED	CBC	128	TTAS.KO-12.0004/R1
	LEA	CBC	128/192/256	TTAK.KO-12.0223
Hash Function (Uni-directional Cryptography) User Data Encryption	SHA-256			ISO/IEC 10118-3
	SHA-384			ISO/IEC 10118-3
	SHA-512			ISO/IEC 10118-3

The TSF performs a cryptographic operation in the cryptographic operation list of [Table 6-8] in accordance with a cryptographic algorithm and a cryptographic key length depending on a component specified in [Table 6-8] while carrying out "cryptographic operation of TSF data."

[Table 6-8] List of Cryptographic Operation Standards

Component	Cryptographic Operation List	Cryptographic Algorithm	Application Mode	Cryptographic Key Length(Bit)	Reference Standard
Management Server	Block Cipher (Symmetrical key cryptography) Encryption/decryption of user data cryptographic keys	ARIA	CBC	256	KS X 1213-1
	Block Cipher (Symmetrical key cryptography) Encryption/decryption of configuration information	ARIA	CBC	256	KS X 1213-1
	Hash function (Uni-directional cryptography) Encryption of administrator password	SHA-2		256	ISO/IEC 10118-3
	Block Cipher (Symmetrical key cryptography) Encryption/decryption of a session cryptographic key	ARIA	CBC	256	KS X 1213-1
	Block Cipher (Symmetrical key cryptography) Encryption/decryption of audit data	ARIA	CBC	256	KS X 1213-1
	Sign Mutual Authentication, TSF Data	RSA-PSS (SHA-256)		2048	PKCS #1 v2.1
	Sign verify Mutual Authentication, TSF Data	RSA-PSS (SHA-256)		2048	ISO/IEC 14888-2
	Integrity verification (TSF data, Encryption of TSF and User Data cryptographic keys)	HMAC-SHA256		256	ISO/IEC 9797-2
	TSF data derived key	PBKDF2(SHA-256)		256	TTAS.KO-12.0334
Agent	Block Cipher (Symmetrical key cryptography) Encryption/decryption of configuration information	ARIA	CBC	256	KS X 1213-1
	Block Cipher (Symmetrical key cryptography) Encryption/decryption of a session cryptographic key	ARIA	CBC	256	KS X 1213-1
	Block Cipher (Symmetrical key cryptography) Encryption/decryption of user data cryptographic keys	ARIA	CBC	256	KS X 1213-1
	Sign	RSA-PSS		2048	PKCS #1 v2.1

	Mutual Authentication, TSF Data	(SHA-256)			
	Sign verify Mutual Authentication, TSF Data	RSA-PSS (SHA-256)		2048	ISO/IEC 14888-2
	Integrity verification (TSF data, Encryption of TSF and User Data cryptographic keys)	HMAC-SHA256		256	ISO/IEC 9797-2
	TSF data derived key	PBKDF2(SHA-256)		256	TTAS.KO-12.0334
Administrative Tool	Block Cipher (Symmetrical key cryptography) Encryption/decryption of configuration information	ARIA	CBC	256	KS X 1213-1
	Block Cipher (Symmetrical key cryptography) Encryption/decryption of a session cryptographic key	ARIA	CBC	256	KS X 1213-1
	Sign Mutual Authentication, TSF Data	RSA-PSS (SHA-256)		2048	PKCS #1 v2.1
	Sign verify Mutual Authentication, TSF Data	RSA-PSS (SHA-256)		2048	ISO/IEC 14888-2
	Integrity verification (TSF data, Encryption of TSF Data cryptographic keys)	HMAC-SHA256		256	ISO/IEC 9797-2
	TSF data derived key	PBKDF2(SHA-256)		256	TTAS.KO-12.0334

The TOE distributes cryptographic keys according to the reference standard in [Table 6-9] by encrypting cryptographic keys of information transmitted among the Management Server, Administrative Tool and Agent with a public key in accordance with the cryptographic algorithm and cryptographic key length specified in [Table 6-9] while performing “distribution of cryptographic keys.”

[Table 6-9] List of Cryptographic Key Distribution Standards

Usage	Cryptographic Algorithm	Cryptographic Key Length	Reference Standard
Public key cryptography	RSAES (SHA-256)	Public key 2048 bits	ISO/IEC 18033-2

While performing “destruction of cryptographic keys”, the TOE destroys memory values of symmetrical keys for the use of KEKs (key Encryption Keys) by overwriting them with 0 during TOE termination, overwrites memory values of symmetrical keys for the use of DEKs (Data Encrypt Keys) during TOE termination and upon the completion of user data encryption, and overwrites cryptographic key files of user data with 0 and deletes the cryptographic key files when the authorized administrator deletes the cryptographic keys using the Administrative Tool.

6.3 Function of User Data Protection

The TOE provides the function of user data encryption/decryption under the encryption/decryption method as per column and does not generate the same cryptogram for the same plain texts.

Using the validated cryptographic module, user data are encrypted depending on the usage such as a block cipher (symmetrical key cryptography) and a hash function (uni-directional cryptography) specified in [Table 6-10] in accordance with a cryptographic algorithm and a cryptographic key length in [Table 6-10] List of Algorithm Standards.

[Table 6-10] List of Algorithm Standards

Usage	Cryptographic Algorithm	Application Mode	Cryptographic Key Length(Bit)	Reference Standard
Block cipher (symmetrical key cryptography) User data encryption/decryption	ARIA	CBC	128/192/256	KS X 1213-1
	SEED	CBC	128	TTAS.KO-12.0004/R1
	LEA	CBC	128/192/256	TTAK.KO-12.0223
Hash function (uni-directional cryptography)	SHA-256			ISO/IEC 10118-3
	SHA-384			ISO/IEC 10118-3

User data encryption	SHA-512			ISO/IEC 10118-3

Using the validated cryptographic module during “TSF data encryption”, the TOE performs encryption of DEKs and configuration files with encryption of cryptographic keys and KEKs in the Agent in accordance with a cryptographic algorithm and a cryptographic key length specified in [Table 6-11] List of Cryptographic Algorithm Standards.

[Table 6-11] List of Cryptographic Key Generation Standards

Cls.	Cryptographic Algorithm	Cryptographic Key Length (Bit)	Reference Standard
Block cipher (Symmetrical key cryptography)	ARIA	256	KS X 1213-1

The TOE generates a random bit from the validated cryptographic module in [Table 6-12], using a random bit generator specified in the following [Table 6-13] List of Random Bit Generation Standards while performing “the function of generating random bits.”

[Table 6-12] Validated Cryptographic Module

Name of Cryptographic Module	Validation No.	Developer	Date Validated
MagicCrypto V2.2.0	CM-162-2025.3	Dream Security	2020-03-03

[Table 6-13] List of Random Bit Generation Standards

Cls.	Random Bit Generation Algorithm	Reference Standard	Remark
Random Bit Generator	HASH_DRBG(SHA-256)	ISO/IEC 18031	Pseudorandom function

While performing encryption/decryption per column specified, the TOE provides users with its capacity of encrypting/decrypting user data through a cryptographic operation in accordance with an algorithm and a cryptographic key length specified in [Table 6-11].

Using the Administrative Tool, the authorized administrator designates columns of the DBMS table protected, establish the application of an algorithm needed and performs encryption/decryption of user data.

If the Administrative Tool requests encryption/decryption of user data for the column established, the Agent performs encryption/decryption with a DEK used for encryption of user data distributed from the Management Server.

Once the authorized administrator completes encryption/decryption of user data, the original user data used are all deleted. If a hash algorithm is used, only encryption can be performed.

Once the authorized administrator completes encryption/decryption of user data, the user data are deleted from the DBMS protected and residual information is destroyed.

Relevant SFR : FCS_CKM.1, FCS_CKM.2, FCS_CKM.4, FCS_COP.1, FCS_RBG.1, FDP_UDE.1, FDP_RIP.1

6.4 Identification and Authentication

If there are [ont to five, (default: five)] unsuccessful authentication attempts by an unauthorized administrator, the TOE shall detect it. In case that the unsuccessful authentication attempts reach to a defined number, the TSF shall inactivate the authentication function for five to ten minutes (default: five minutes), deny any authentication and send a warning email to the authorized administrator.

In the TOE, the identification and authentication of the administrator are performed at once. The information provided through the screen GUI for identification/authentication of the administrator is an ID and a password, which are used to identify/authenticate the administrator. An action that can be taken before the administrator is identified/authenticated is a communication check. The administrator manages the security functions after he or she is successfully identified/authenticated.

The TOE performs mutual authentication through a self-implemented authentication protocol between the Management Server and Agent.

The Management Server performs mutual authentication of users between the Agent and management tool though a self-implemented authentication protocol based on a certificate, ID and password.

1. The Agent or Administrative Tool conducts mutual authentication by installing their certificates respectively issued from the certificate of the Management Server in advance.
2. Once the Agent or Administrative Tool is accessed to the Management Server, the server signs on (RSA-PSS(SHA-256)) with a private key of the Management Server, generates SignedData (PKCS#7) including the certificate of the server and sends it to the Agent or Administrative Tool.
3. The Agent or Administrative Tool performs a chain verification on the certificate of the Management Server included in the SignedData received and a certificate of the Agent or Administrative Tool.
4. The Agent and Administrative Tool generate random Key (1) and IV (1) (HASH_DRBG(SHA-256)).
5. SignedData are generated using a private key of the Agent or Administrative Tool including cryptographic data encrypted (RSAES(SHA-256)) with the certificate of the Management Server that is the recipient of Key (1) and IV (1) randomly generated, and the SignedData generated are transmitted to the Management Server.
6. The Management Server performs mutual authentication by validating and verifying a chain of the Agent or Administrative Tool's certificate.
7. The Key (1) and IV (1) encrypted in the SignedData that are received from the Agent or Administrative Tool are decrypted with a private key of the Management Server.
8. The Management Server generates Key (2) and IV (2) randomly (HASH_DRBG(SHA-256)).
9. The Key (2) and IV (2) randomly generated are encrypted (RSAES(SHA-256)), using the certificate of the Agent or Administrative Tool received, and are transmitted to the Agent or Administrative Tool.
10. The SignedData including Key (2) and IV (2) encrypted (RSAES(SHA-256)), using the private key of the Agent or Administrative Tool received, are transmitted to the Agent or Administrative Tool.
11. The Agent or Administrative Tool decrypts cryptographic Key (2) and IV(2) received, using a private key of the Agent or Administrative Tool, and relies on cryptographic communications based on a session key for communication afterwards.

The TOE shall satisfy password rules and provide a mechanism to verity that the password rules such as a combination of all three character types, namely, valid English characters (a-z, A~Z), numbers (0~9) and special characters (~, `, !, @, #, $, %, ^, &, *, (,), -, _, +, =) in a length of 9 to 16 digits are met.
The password is verified during the Administrative Tool's login to the Management Server and the start-up of the Agent and the Administrative Tool. f password shall be verified whenever
The TOE shall allow establishment of Management Server IP and Port, exchanges of Nonce values and agreement of session keys to be enforced on behalf of an administrator before the administrator is authenticated. The administrator shall be successfully authenticated prior to permission of all other actions mediated by the TSF on behalf of the administrator.

The TOE shall prevent re-use of authentication data related to user authentication.

1. The Administrative Tool or Agent generates Nonce (R1) and transmits the ID and Nonce (R1) to the Management Server.
2. The server generates Nonce (R2), and performs an XOR operation of Nonce (R1), Nonce (R2)

and PWD (1) after viewing the PWD (1) and Salt (1) with the subject ID of the Administrative Tool or Agent subject ID, followed by a hash (SHA256) to generate AuthValue (A1).

3. The Salt (1) viewed using the subject ID of the Administrative Tool or Agent and Nonce (R2) generated by the Management Server are transmitted to the Agent.
4. The Administrative Tool or Agent generates PWD (2) by associating Password (1) with Salt (1) received from the Management Server.
5. Nonce (R2) received and PWD (2) that previously generated Nonce (R1) are used to perform an XOR operation and hash to create AuthValue (A2). The AuthValue (A2) generated is transmitted to the Management Server.
6. The Management Server verifies AuthValue (A2) received by the Agent by comparing it with AuthValue (A1). The Agent performs authentication and transmits answers including the outcome of a login request if the authentication succeeds and a new session key in case of login success.
7. Re-use of authentication data is prevented by comparing AuthValue (A1) and AuthValue (A2).

A. With the Administrative Tool that uses a password-based authentication method, the TOE receives an ID and password and allows the authorized administrator to manage security functions.

B. An action that can be performed before an administrator is authenticated includes establishment of Management Server IP and port and, after the authentication, the administrator can manage security functions.

C. The Management Server performs mutual authentication of users between the Agent and Administrative Tool through a self-implemented protocol based on a certificate, ID and password.

1. The Agent or Administrative Tool conducts mutual authentication by installing their certificates respectively issued from the certificate of the Management Server in advance.
2. Once the Agent or Administrative Tool is accessed to the Management Server, the server signs on (RSA-PSS(SHA-256)) with a private key of the Management Server, generates SignedData (PKCS#7) including the certificate of the server and sends it to the Agent or Administrative Tool.
3. The Agent or Administrative Tool performs a chain verification on the certificate of the Management Server included in the SignedData received and a certificate of the Agent or Administrative Tool.
4. The Agent and Administrative Tool generate random Key (1) and IV (1) (HASH_DRBG(SHA-256)).
5. SignedData are generated using a private key of the Agent or Administrative Tool including cryptographic data encrypted (RSAES(SHA-256)) with the certificate of the Management Server that is the recipient of Key (1) and IV (1) randomly generated, and the SignedData generated are transmitted to the Management Server.
6. The Management Server performs mutual authentication by validating and verifying a chain of the Agent or Administrative Tool's certificate.
7. The Key (1) and IV (1) encrypted in the SignedData that are received from the Agent or Administrative Tool are decrypted with a private key of the Management Server.
8. The Management Server generates Key (2) and IV (2) randomly (HASH_DRBG(SHA-256)).
9. The Key (2) and IV (2) randomly generated are encrypted (RSAES(SHA-256)), using the certificate of the Agent or Administrative Tool received, and are transmitted to the Agent or Administrative Tool.

10. The SignedData including Key (2) and IV (2) encrypted (RSAES(SHA-256)), using the private key of the Agent or Administrative Tool received, are transmitted to the Agent or Administrative Tool.
11. The Agent or Administrative Tool decrypts cryptographic Key (2) and IV(2) received, using a private key of the Agent or Administrative Tool, and relies on cryptographic communications based on a session key for communications afterwards.

The TOE masks with * characters revealed to users while the administrator enters the password and provides feedback in the following texts in case of authentication failure with no mention of a failure reason: "Administrator has failed to log in."

The TOE shall allow establishment of the Management Server IP and port to be enforced on behalf of the authorized administrator before the administrator is identified, and successfully identify each authorized administrator prior to permission of all other actions mediated by the TSF on behalf of the authorized administrator. The Agent does not allow any additional IP.

Relevant SFR : FIA_AFL.1, FIA_IMA.1, FIA_SOS.1, FIA_SOS.2, FIA_SOS.3, FIA_UAU.1, FIA_UAU.4, FIA_UAU.7, FIA_UID.1,

6.5 Security Management

The TOE restricts management actions to the authorized administrator with the capacity of managing security functions specified in [Table 6-14].

[Table 6-14] List of Authorized Administrator's Security Functions

Security Function	Management Action
Set-up and modification of administrator password	Determine and modify an action
Set-up and modification of audit information thresholds	Determine and modify an action
Modification of the number of authentication failures and inactivity time	Determine and modify an action
Set-up and modification of the mail server	Determine and modify an action
Set-up and modification of administrator access IP	Determine and modify an action
Real-time validation of modules	Initiate an action
View of audit information	Determine an action
Set-up and modification of cryptographic keys	Determine and modify an action
Establishment and modification of policies	Determine and modify an action
Column encryption	Determine an action

The TOE restricts its capacity of managing TSF data specified in [Table 6-15] to the authorized administrator.

[Table 6-15] List of Authorized Administrator's TSF Data

TSF Data List	Management
Audit information	Query
Modification of administrator password	Modification
Set-up of audit thresholds	Query, modification
Set-up of the mail server	Query, modification
Administrator information	Query, modification
Set-up of administrator access IP	Query, modification
Cryptographic keys (user data encryption)	Query
Policy list	Query, modification
Authentication failure information	Query, modification
Column list	Query, modification

The TOE provides a function of changing the password when the authorized administrator accesses for the first time. The password is defined based on rules such as a combination of the three character types, namely, valid English characters (a-z, A~Z), numbers (0~9) and special characters (~, `, !, @, #, $, %, ^, &, *, (,), -, _, +, =) in a

valid length of 9 to 16 digits. The authority over creating and changing an ID and a password for the administrator and the Agent is restricted only to the authorized administrator.
The authorized administrator of the TOE plays a role in setting and changing the administrator password, setting and changing an audit threshold, changing the number of authentication failure and inactivity time, setting and changing the mail server, setting and changing administrator access IP, validating modules in real time, viewing audit information, setting and changing cryptographic keys, setting and changing policies and encrypting columns.

Relevant SFR : FMT_MOF.1, FMT_MTD.1, FMT_PWD.1, FMT_SMF.1, FMT_SMR.1,

6.6 Protection of the TSF

The TOE protects data from disposure and modification through encryption (ARIA-128-CBC) of a validated cryptographic module during data transmission between the Management Server and Agent.

1. The Administrative Tool conducts mutual authentication by installing its certificate issued from the certificate of the Management Server in advance.
2. Once the Administrative Tool is accessed to the Management Server, the server signs on (RSA-PSS(SHA-256)) with a private key of the Management Server, generates SignedData (PKCS#7) including the certificate of the server and sends it to the Administrative Tool.
3. The Administrative Tool performs a chain verification on the certificate of the Management Server included in the SignedData received and the Administrative Tool's certificate.
4. The Administrative Tool generates random Key (1) and IV (1) (HASH_DRBG(SHA-256)).
5. SignedData are generated using a private key of the Administrative Tool including cryptographic data encrypted (RSAES(SHA-256)) with the certificate of the Management Server that is recipient of Key (1) and IV (1) randomly generated, and the SignedData generated are transmitted to the Management Server.
6. The Management Server performs mutual authentication by validating and verifying a chain of the Administrative Tool's certificated.
7. The Key (1) and IV (1) encrypted in SignedData that are received from the Administrative Tool are decrypted with a private key of the Management Server.
8. The Management Server generates Key (2) and IV (2) randomly (HASH_DRBG(SHA-256)).
9. The SignedData including Key (2) and IV (2) encrypted (RSAES(SHA-256)), using the certificate of the Administrative Tool that received Key (2) and IV (2) randomly generated, are transmitted to the Administrative Tool.
10. The Administrative Tool decrypts cryptographic Key (2) and IV (2) received, using a private key of the Administrative Tool, and relies on cryptographic communications (ARIA-128-CBC) with a session key for communication afterwards.

With a validated cryptographic module that performs encryption, the TOE protects against unauthorized disclosure and modification.

[Table 6-16] Methods of TSF Data Protection

Component	To be Encrypted	Cryptographic Key	Cryptographic Algorithm	Storage Location
Management Server	Cryptographic key of user data	Key generated (DEK)	ARIA/CBC/256	File
	Configuration information	Key generated (DEK)	ARIA/CBC/256	File
	Administrator password	Random value (SALT)	SHA256	File

	Session cryptographic key	Key generated (DEK)	ARIA/CBC/256	File
	Critical security parameter	Key generated (DEK)	ARIA/CBC/256	File
	Audit data	Key generated (DEK)	ARIA/CBC/256	File
	Validated cryptographic module	-	SHA256	File
	Executable script	-	SHA256	File
	Log file	-	SHA256	File
Agent	Configuration information	Key generated (DEK)	ARIA/CBC/256	File
	Session cryptographic key	Key generated (DEK)	ARIA/CBC/256	Memory
	Cryptographic key of user data	Key generated (DEK)	ARIA/CBC/256	Memory
	Critical security parameter	Key generated (DEK)	ARIA/CBC/256	Memory
	Validated cryptographic module	-	SHA256	File
	Executable script	-	SHA256	File
	Log file	-	SHA256	File
Administrative Tool	Configuration information	Key generated (DEK)	ARIA/CBC/256	File
	Validated cryptographic module	-	SHA256	File
	Session cryptographic key	Key generated (DEK)	ARIA/CBC/256	Memory

User data cryptographic keys of the Management Server randomly (HASH_DRBG(SHA-256)) generate a DEK and IV for a DEK to encrypt the user data cryptographic keys when a cryptographic key is generated. The DEK and IV for DEK randomly generated encrypt user data cryptographic keys (ARIA/CBC/256). The DEK and IV for DEK that utilize user data cryptographic keys for encryption are encrypted (ARIA/CBC/256) with a KEK and stored in a separate file. The IV for DEK used herein is an IV for a KEK randomly generated as per user data cryptographic key in the file DB stored along with critical security parameters. User data cryptographic keys encrypted (ARIA/CBC/256) and the DEK and IV for DEK that encrypt (ARIA/CBC/256) user data cryptographic keys ensure integrity with HMAC-SHA256 by using the KEK as a key.

The Agent performs encryption (ARIA/CBC/256) with a cryptographic key (DEK and IV) shared from mutual authentication of user data cryptographic keys generated in the Management Server, to be delivered in real time during start-up of the Agent or generation of user data cryptographic keys, while the user data cryptographic keys delivered are stored in shared memories that keep user data cryptographic keys, policy names and critical security parameters of user data encryption that have been encrypted (ARIA/CBC/256) with the DEK and IV for DEK randomly generated (HASH_DRBG(SHA-256)). The DEK and IV for DEK are encrypted (ARIA/CBC/256) with a KEK while IV for KEK is randomly generated (HASH_DRBG(SHA-256) and stored in shared memories.

The configuration files used in the TOE are comprised of DEKs and IV for DEKs randomly generated (HASH_DRBG(SHA-256)) which encrypt (ARIA/CBC/256) configuration detail when the configuration files are created. The DEKs and IV for DEKs are encrypted(ARIA/CBC/256) into KEKs and IV for KEKs to store encrypted configuration detail in the configuration files that store Slat used for generation of a KEK, encrypted DEKs and IV for DEKs following the configuration detail encrypted and cryptographic module files and a hash of each executable shell script during installation. All details stored ensure integrity based on HMAC-SHA256 using the KEK as a key.

The section between the Management Server and Administrative Tool and that between the Management Server and Agent of the TOE are connected through an encryption session. An encryption session key for the encryption session is distributed through a self-implemented mutual authentication protocol, using a certificate. When a session key is generated or distributed, the encryption session key randomly (HASH_DRBG(SHA-256)) generates a DEK for encryption (ARIA/CBC/256) to be encrypted (ARIA/CBC/256) with a KEK and IV

for a KEK and stored.

The IV for DEK, critical security parameters and audit data needed for encryption of user data cryptographic key files in the management files are encrypted and stored in the file DB which is comprised of multiple pages in a unit of 8192 bytes. On each page, critical TSF data such as critical parameters and audit data are encrypted and stored as a DEK and IV for a DEK randomly generated (HASH_DRBG(SHA-256)). The IV for KEK and encrypted DEK and IV for DEK are stored on each page. Integrity is ensured based on HMAC-SHA256 using the KEK as a key.

The Management Server and Agent generate a log file for tracking of execution and the log file ensures integrity as per each line based on a hash (SHA-256). The hash of each line is associated with the hash value of the previous line, generating association among the previous, current and next lines and ensuring integrity. As for the first line, a hash value is generated in association with a KEK. The integrity of the entire log file is ensured by verifying the first line whose hash value is associated with the KEK, followed by subsequent lines in association with the hash value of the previous line.

The KEK induces a key based on PBKDF2(SHA-256) algorithm of a validated cryptographic module, using Salt (16 bytes) and iteration counts (1024) stored in the configuration file and password entered during the start-up. It performs an XOR operation along with a vector (32 bytes) which is an additional random generation (HASH_DRBG(SHA-256)) and is encoded to Base64. The password entered is overwritten with 0 and destroyed.

A self-test is performed to maintain secure and accurate operation conditions during the start-up of components and normal operation on a periodical basis, and upon the request of the administrator, using a TSF data protection method specified in [Table 6-16] for protection of the TSF.
The self-test is conducted during the initial start-up or every three hours after loading.
The self-test is conducted on test items in [Table 6-17] depending on the TOE classification in [Table 6-17]; and in case of a self-test failure, a warning message is notified to the administrator via email along with details including the time of self-test failure (reliable OS time of the operational environment) and the failed file(s).

[Table 6-17] Items of Self-test

TOE Classification	Self-test Item	Testing
Management Server	Validated Cryptographic Module MagicCrypto V2.2.0	Self-test internally performed by the validated cryptographic module
	Process Name MagicDBPolicy	Check if main processes needed to run the Management Server are in normal operation and send the result to audit logs
Agent	Validated Cryptographic Module MagicCrypto V2.2.0	Self-test internally performed by the validated cryptographic module
	Process Name MDBAgent	Check if main processes needed to run the Agent are in normal operation and send the result to audit logs
Administrative Tool	Validated Cryptographic Module MagicCrypto V2.2.0	Self-test internally performed by the validated cryptographic module
	Process Name MagicDBPlus_v2.0_Admin	Check if main processes needed to run the Administrative Tool are in normal operation and send the result to audit logs

The following [Table 6-18] describes TSF integrity test methods under which integrity is tested for test items in [Table 6-18] depending on the TOE classification in [Table 6-18] in accordance with the testing detail specified in [Table 6-18]. In case integrity verification fails due to data damage from an unauthorized user, a warning message is sent to the administrator via email.

[Table 6-18] Items of TSF Integrity Test

TOE Classification	Item of Integrity Test	Testing
Management Server	MagicDBPolicy.conf	The entire configuration information is verified with a signature value using the certificate issued during the initial installation, and HMAC generated with a KEK is stored when the configuration file is created. An integrity test is performed to see if there is any falsification due to an unauthorized user

TOE Classification	Item of Integrity Test	Testing
		during the start-up of the Management Server, upon the request of the administrator using the Administrative Tool and every three hours since the start-up of the Management Server. The test results are transmitted to audit logs and, if falsified, notified to the administrator via email together with the name of a falsified file. In case of configuration damage that keeps the Management Server from being in normal operation, the audit log and email are sent during the server's next normal operation.
	libMagicCrypto.so start.sh stop.sh restart.sh	Hash (SHA256) values from libMagicCrypto.so, start.sh, stop.sh, restart.sh installed during generation of the configuration file are stored in the configuration file. An integrity test is performed to see if there is any falsification due to an unauthorized user during the start-up of the Management Server, upon the request of the administrator using the Administrative Tool and every three hours since the start-up of the Management Server. The test results are transmitted to audit logs and, if falsified, notified to the administrator via email together with the name of a falsified file. In case of the cryptographic module's damage that keeps the Management Server from being in normal operation, the audit log and email are sent during the server's next normal operation.
	magicdb.dat magicdb.audit.dat	HMAC generated with a KEK for each block in a certain size is stored during data recording. An integrity test is performed to see if there is any falsification due to an unauthorized user during the start-up of the Management Server, upon the request of the administrator using the Administrative Tool, and every three hours since the start-up of the Management Server. The test results are transmitted to audit log and, if falsified, notified to the administrator via email together with the name of a falsified file. In case of file DB damage that keeps the administrator's email address and email server information from being successfully uploaded, the email is sent, using the information in the memories stored by the Management Server during the start-up and a change in information of the administrator's email address and email server.
	Cryptographic Key File	HMAC generated with a KEK is stored in the file when a cryptographic key is generated. An integrity test is performed to see if there is any falsification due to an unauthorized user during the start-up of the Management Server, upon the request of the administrator using the Administrative Tool and every three hours since the start-up of the Management Server. The test results are transmitted to audit log and, if falsified, notified to the administrator via email together with the name of a falsified file
	Log File	If a log is added to the log file, a chain verification is needed for each line of logs using a hash (SHA256). When the initial log file is generated, its first line and a KEK are associated together to generate a hash value which is recorded on the last line of the log. The following line is associated together with the previous line to generate a hash value. The entire file can be verified by associating the first line with a KEK and performing a hash operation to compare and verify the hash value recorded on the corresponding line, followed by the subsequent lines which go through a hash operation and verification. An integrity test is performed to see if there is any falsification due to an unauthorized user during the start-up of the Management Server, upon the request of the administrator using the Administrative Tool and every three hours since the start-up of the Management Server. The test results are transmitted to audit logs and, if falsified, notified to the administrator via email together with the name of a falsified file.

TOE Classification	Item of Integrity Test	Testing
Agent	MDBAgent.conf	The entire configuration information is verified with a signature value using the certificate issued during the initial installation, and HMAC generated with a KEK is stored when the configuration file is created. An integrity test is performed to see if there is any falsification due to an unauthorized user during the start-up of the Management Server, upon the request of the administrator using the Administrative Tool and every three hours since the start-up of the Management Server. The test results are transmitted to audit logs and, if falsified, notified to the administrator via email together with the name of a falsified file. In case of configuration damage that keeps the Agent from being in normal operation, the audit log and email are sent during next normal operation.
	libMagicDB.so libMagicCrypto.so start.sh stop.sh restart.sh	Hash (SHA256) values from libMagicDB.so, libMagicCrypto.so, start.sh, stop.sh, restart.sh installed during generation of the configuration file are stored in the configuration file. An integrity test is performed to see if there is any falsification due to an unauthorized user during the start-up of the Management Server, upon the request of the administrator using the Administrative Tool and every three hours since the start-up of the Management Server. The test results are transmitted to audit logs and, if falsified, notified to the administrator via email together with the name of a falsified file. In case of the cryptographic module's damage that keeps the Agent from being in normal operation, the audit log and email are sent during next normal operation.
	Log File	If a log is added to the log file, a chain verification is needed for each line of logs using a hash (SHA256). When the initial log file is generated, its first line and a KEK are associated together to generate a hash value which is recorded on the last line of the log. The following line is associated together with the previous line to generate a hash value. The entire file can be verified by associating the first line with a KEK and performing a hash operation to compare and verify the hash value recorded on the corresponding line, followed by the subsequent lines which go through a hash operation and verification. An integrity test is performed to see if there is any falsification due to an unauthorized user during the start-up of the Management Server, upon the request of the administrator using the Administrative Tool and every three hours since the start-up of the Management Server. The test results are transmitted to audit logs and, if falsified, notified to the administrator via email together with the name of a falsified file.
	DEK	A DEK for user data delivered to the Management Server is encrypted (ARIA-256-CBC) and stored in the memories and, while being stored, its integrity is ensured using HMAC values generated with a KEK. An integrity test is performed to see if there is any falsification due to an unauthorized user during the start-up of the Management Server, upon the request of the administrator using the Administrative Tool and every three hours since the start-up of the Management Server. The test results are sent to the administrator via email.
Administrative Tool	MDBAdmin.xml.enc	HMAC generated with a KEK is stored when the configuration file is created. An integrity test is performed to see if there is any falsification due to an unauthorized user during the start-up of the Management Server, upon the request of the administrator using the Administrative Tool and every three hours since the start-up of the Management Server. The test results are transmitted to audit logs and, if falsified, notified to the administrator via email together with the name of a falsified file. In case of configuration damage that keeps the Administrative Tool from being in normal operation, the audit

TOE Classification	Item of Integrity Test	Testing
		log and email are sent during next normal operation.
	MagicCryptoV22.dll	Hash (SHA256) values from MagicCryptoV22.dll, install.ico, uninstall.ico, Uninstall.exe installed during generation of the configuration file are stored in the configuration file. An integrity test is performed to see if there is any falsification due to an unauthorized user during the start-up of the Management Server, upon the request of the administrator using the Administrative Tool, and every three hours since the start-up of the Management Server. The test results are transmitted to audit logs and, if falsified, notified to the administrator via email together with the name of a falsified file. In case of the cryptographic module's damage that keeps the Administrative Tool from being in normal operation, the audit log and email are sent during next normal operation.
	install.ico	
	uninstall.ico	
	Uninstall.exe	

Relevant SFR : FPT_ITT.1, PST_ITT.1, FPT_TST.1,FPT_PST.1

6.7 TOE Access

The TOE limits the maximum number of concurrent sessions of management access by the same administrator with the same authority to one. If there is another attempt to access using the same account after the administrator has logged in, the first comer takes the priority and the new access is blocked.

The TOE is only accessible from the Administrative Tool with a pre-registered IP address, and as for the management access, any access not from the access IP defined by the authorized administrator shall be denied. By default, there are two access IPs defined by the authorized administrator and they can be added or deleted using the Administrative Tool.

- The access IP is restricted to IPv4's address system.
- The IPv4's address system is comprised of 12 digits in total which can be divided into four sections with three digits per each. Each section is expressed as an integer of 1 to 255 in one to three digits and is divided with '.' as in A.B.C.D.

Once the administrator has logged in, the access session performs session termination after inactivity time without any action for a certain period. The administrator's inactivity time is defined as 10 minutes and the absence of following requests that communicate with the Management Server is considered as inactivity time:

- Request to view a list of cryptographic keys
- Request to register a cryptographic key
- Request to modify a cryptographic key
- Request to search a policy list
- Request to register a policy
- Request to view a list of users
- Request to register a user
- Request to modify a user
- Request to search logs

Relevant SFR : FTA_MCS.2, FTA_SSL.2, FTA_TSE.2, FTA_SSL.5, FTA_TSE.1