Security Target (ST)

- D'Guard v5.0 -

Version 1.2

The Security Target is related to the certified TOE. This Security Target is written in Korean and translated from Korean into English.

Revision History

Ver	Date	Revision	Author
1.0	May 14, 2024	First Issue	Jaehong Kang
1.1	Jul 12, 2024	TOE Requirements, etc. to take effect	Jaehong Kang
1.2	Sep 17, 2024	TOE Version, etc. to take effect	Jaehong Kang

Table of Contents

1. ST Introduction

This chapter introduces the Security Target (ST) of D’Guard v5.0 of INEB Inc.

1.1. ST reference

[Table 1] ST reference

Title	D’Guard v5.0 Security Target
Version	V1.2
Evaluation Assurance Level	EAL1+(ATE_FUN.1)
Developer	INEB Inc.
Common Criteria	Common Criteria for Information Technology Security Evaluation (Notification No. 2013-51 of the Ministry of Science, ICT and Future Planning) Common Criteria for Information Technology Security Evaluation · Common Criteria Part 1 : Introduction and General Model v3.1 r5 (Version 3.1 revision 5, April 2017, CCMB-2017-04-001) · Common Criteria Part 2 : Security Functional Components v3.1 r5 (Version 3.1 revision 5, April 2017, CCMB-2017-04-002) · Common Criteria Part 3 : Security Assurance Components v3.1 r5 (Version 3.1 revision 5, April 2017, CCMB-2017-04-003)
Protection Profile	Korean National Protection Profile for Database Encryption V1.1, KECS-PP-0820a-2017, December 11, 2019
Common Criteria Version	CC V3.1 r5
Product Classification	DB Encryption
Keywords	Database, Encryption
Issue Date	Aug 19, 2024

1.2. TOE reference

[Table 2] TOE reference

TOE Identification	D’Guard v5.0
TOE Version	V5.0 (v5.0.1)
TOE Component	D’Guard KMS v5.0.1 D’Guard Plug-In v5.0.1 D’Guard API v5.0.1
Final Release	August 19, 2024
TOE Developer	INEB Inc.

1.3. TOE overview

1.3.1. Database encryption overview

D’Guard v5.0 (hereinafter referred to as the TOE) is the product that encrypts important information in the database (hereinafter “DB”) by the unit of column to contain the confidential information. The TOE performs the function of preventing the unauthorized disclosure of confidential information. The encryption target of the TOE is the DB managed by the database management system (hereinafter “DBMS”) in the organization’s operating environment, and some or all user data is encrypted according to the security policy. The main security functions provided by the TOE are the encryption and decryption of data stored in the DB, cryptographic key management and auditing.

1.3.2. TOE type and scope

D’Guard v5.0 (hereinafter referred to as the TOE) is database encryption software that encrypts and decrypts the user data in a column of a database to be protected.

The TOE types defined in this ST are grouped into the ‘plug-in type’ and ‘API type’, depending on the TOE operation type. The TOE supports both types. The plug-in type is a method that performs encryption on the DB Server and operates dependent on the database. Since TOE module(the plug-in type) is installed in the protected DB server to perform encryption/decryption, the DBMS is limited to Oracle database and Tibero database. The API type performs encryption in the application server (hereinafter 'AP') and is classified into Java API according to application work environment.

Components of the TOE are the management server that manages policies and keys (hereinafter referred to as D'Guard KMS), agents that perform encryption in the Java work environment (hereinafter referred to ad D'Guard API). And it consists of agent (D'Guard Plug-In) which performs encryption in DB Server. The D'Guard KMS provides services that perform key distribution roles, perform policy management roles and collect agent logs. The TOE administrator accesses the D'Guard KMS through a web browser and performs management roles. It also plays an administrative role in the console environment for initialization.

1.3.3. TOE usage and major security features

The TOE is used to encrypt/decrypt the user data stored in the DB server according to the policy set by the authorized administrator to prevent the unauthorized disclosure of the confidential information.

In order that the authorized administrator can operate the TOE securely in the operational environment of the organization, the TOE provides various security features such as the security audit function that records and manages major auditable events; cryptographic support function such as cryptographic key management to encrypt the user and the TSF data, and cryptographic operation; user data protection function that encrypts the user data and protects the residual information; identification and authentication function such as verifying the identity of the authorized administrator, authentication failure handling, and mutual authentication among the TOE components; security management function for security functions, role definition, and configuration; TSF protection functions including protecting the TSF data transmitted among the TOE components, protecting the TSF data stored in the storage that is controlled by the TSF, and TSF self-test; and TOE access function to manage the access session of the authorized administrator.

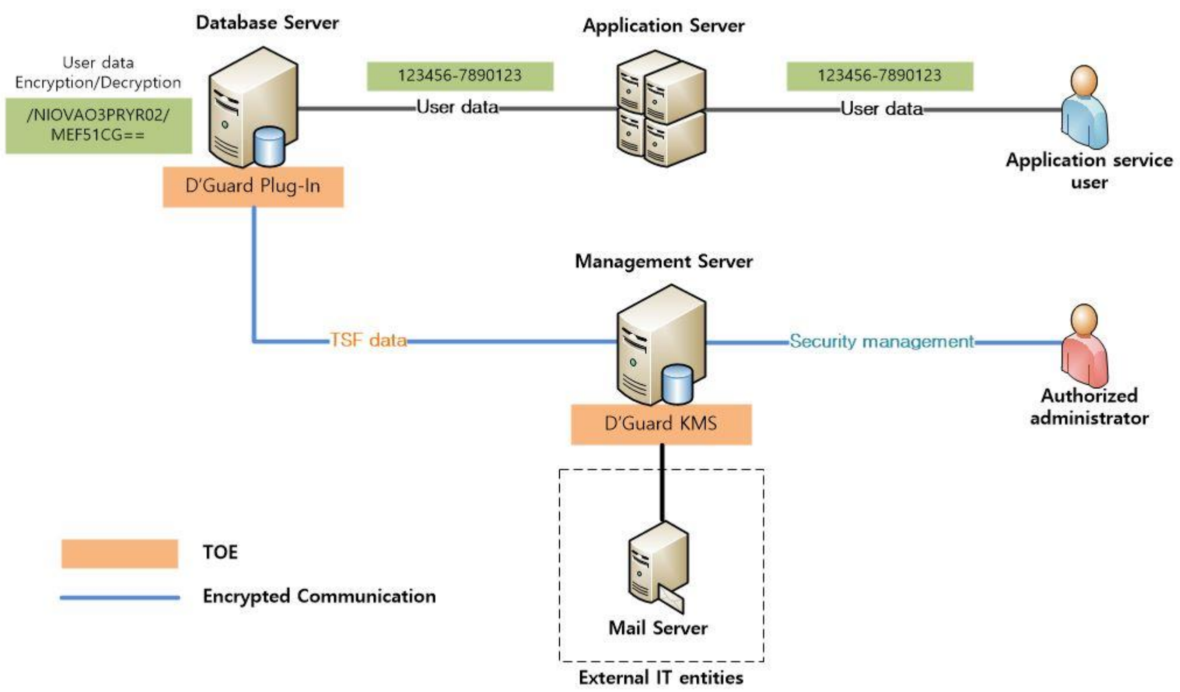
The DEK (Data Encryption Key) used to encrypt/decrypt the user data is protected by encryption with the KEK (Key Encryption Key). For the requirements regarding how to generate and use the DEK and KEK, refer to 5.1.2 Cryptographic Support (FCS).

1.3.4. Non-TOE and TOE operational environment

The TOE operational environment defined in this ST can be classified into two: plug-in type and API type.

[Figure 1] shows the general operational environment of the plug-in type. The agent, which is installed in the protected database server of the DB, encrypts the user data received from the application server before storing it in the DB according to the policy configured by the authorized administration, and decrypts the encrypted user data sent form the database server to the application server.

Since the TOE module(the plug-in type) is installed in the protected database to perform encryption/ decryption, the DBMS that can be installed and operated by the TOE is limited to Oracle 19.3 and Tibero 7.

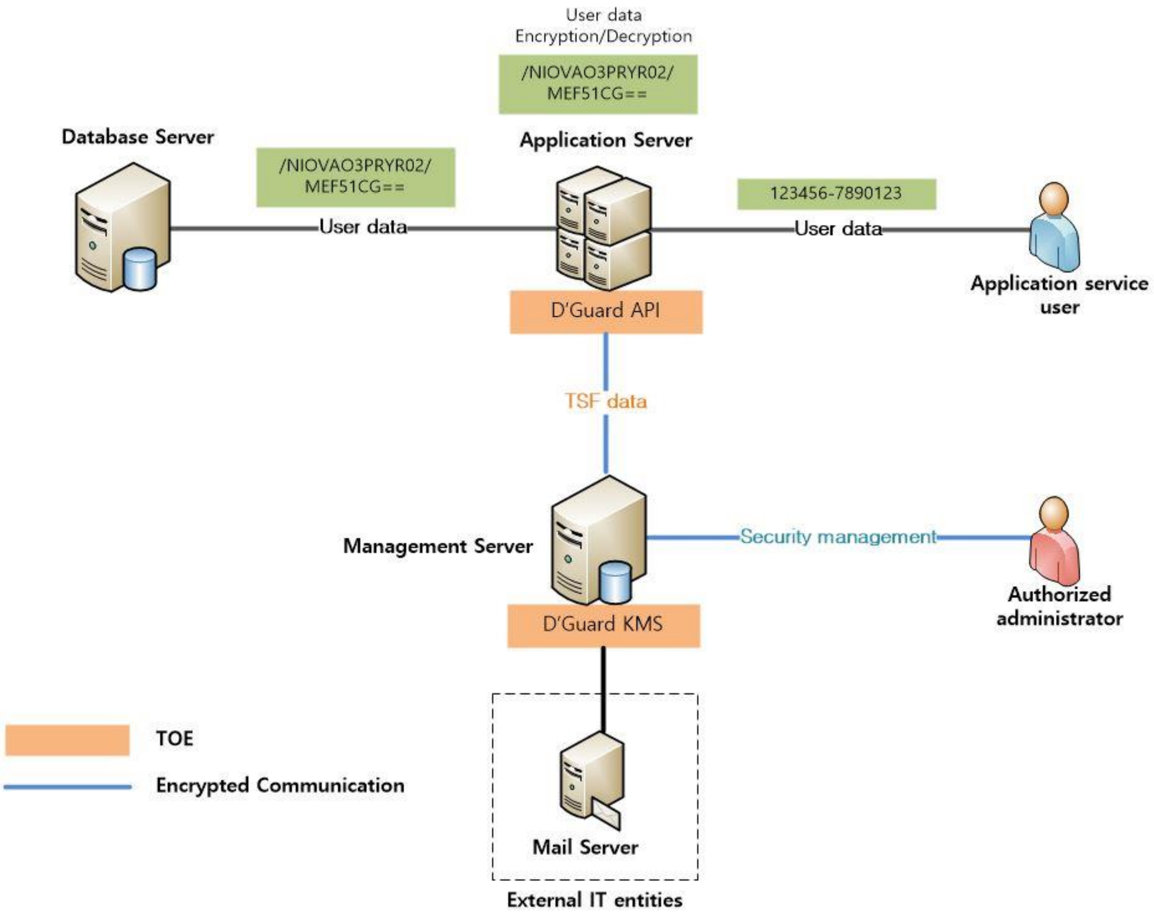


[Figure 1] Plug-In type operational environment

The authorized administrator performs policy management distributed to D’Guard Plug-In according to the scope required by the organizations security policy through D’Guard KMS. In addition, the authorized administrator can perform security management through access to the management server.

[Figure 2] shows the general operational environment of the API type. The application, which is installed in the application server and provides application services, is developed using the D’Guard

API provided by API module in order to use the cryptographic function of the TOE. The D'Guard API module is installed in the application server and performs encryption/decryption of the user data in accordance with the policies configured by authorized administrator. The user data entered by the application service user is encrypted by D'Guard API module, which is installed in the application server, and sent to the database server. The encrypted user data received from the database server is decrypted by the D'Guard API module, which is installed in the application server, and sent to the application service user.



[Figure 2] API type operational environment

The authorized administrator performs policy management through the D'Guard KMS as an D'Guard API module according to the scope of the organization's security policy. In addition, the authorized administrator can perform security management through access to the management server.

The communication among the TOE components performs encrypted communication using the approved cryptographic algorithm of the validated cryptographic module, and the transmitted the TSF data includes security policy data and audit data transmitted from the agent. The self-implemented mutual authentication is performed when communicating among the TOE components.

Administrator accesses the management server through a web browser to perform security management functions. HTTPS (TLS 1.2), which implements a secure security protocol, is used for the communication section of the web environment-based administrator. The communication section for management access is excluded from the evaluation scope.

As an external IT entity required to operate TOE, there is a Mail server for administrator notifications authorized when predicting audit data loss. TOE works with an external Mail server to send security alerts to administrators according to security policies defined by the administrator.

The hardware and software requirements for operating the TOE are as follows.

[Table 3] Requirements for TOE operational environment of the D’Guard KMS

D’Guard KMS			
HW (or SW)	Type		Minimum operation specification
Hardware	CPU		Intel® Core™ i5 CPU @ 3.00 GHz or higher
	RAM		8GB or higher
	HDD		10GB or higher necessary for installing the TOE
	NIC		100/1000 Mbps 1 network port or higher
Software	OS		Ubuntu 20.04 (Linux Kernel 5.4.0-181) 64bit
	3rd Party	DBMS	PostgreSQL 15.8
		WAS	Apache Tomcat 9.0.95
		Java	JRE 8(Java SE Runtime Environment 8u421)

OS (Ubuntu 20.04)

[Tbale3] shows the operating system for the D’Guard KMS operation.

DBMS (PostgreSQL 15.8)

The DBMS provides the basic storage function of the D’Guard KMS to store policy information registered by administrator and log information collected by each agent.

WAS (Apache Tomcat 9.0.95)

It is included with the D’Guard KMS and installed when installing management server. It provides web-based security management interface (GUI : Graphical User Interface) through WAS.

JRE 8(Java SE Runtime Environment 8u421)

It is the foundation framework for running the D’Guard KMS’s security management interface.

[Table 4] Requirements for TOE operational environment of the D’Guard Plug-In

D’Guard Plug-In		
HW (or SW)	Type	Minimum operation specification
Hardware	CPU	Intel® Core™ i5 CPU @ 3.00 GHz or higher
	RAM	8GB or higher
	HDD	1GB or higher necessary for installing the TOE
	NIC	100/1000 Mbps 1 network port or higher
Software	OS	Rocky 8.6 (Linux Kernel 4.18.0-372) 64bit
	DBMS	Oracle 19.3
		Tibero 7

OS (Rocky 8.6)

[Tbale3] shows the operating system for the D’Guard Plug-In operation.

DBMS (Oracle 19.3)

DBMS is operational environment Oracle database which the D’Guard Plug-In is installed

DBMS (Tibero 7)

DBMS is operational environment Tibero database which the D’Guard Plug-In is installed

[Table 5] Requirements for TOE operational environment of the D’Guard API

D’Guard API		
HW (or SW)	Type	Minimum operation specification
Hardware	CPU	Intel® Core™ i5 CPU @ 3.00 GHz or higher
	RAM	8GB or higher
	HDD	1GB or higher necessary for installing the TOE
	NIC	100/1000 Mbps 1 network port or higher
Software	OS	Ubuntu 20.04 (Linux Kernel 5.4.0-181) 64bit
	Java	JRE 8(Java SE Runtime Environment 8u421)

OS (Ubuntu 20.04)

[Tbale3] shows the operating system for the D’Guard API operation.

JRE 8(Java SE Runtime Environment 8u421)

It is a work program development environment which the D’Guard API is installed.

[Table 6] Requirements for TOE operational environment of the administrator PC

TOE administrator PC		
HW (or SW)	Type	Minimum operation specification
Software	Web Browser	Chrome 129.0 (64bit)

Web Browser (Chrome 1290)

It provides GUI of web environment to execute manager of the D'Guard KMS.

External IT entities

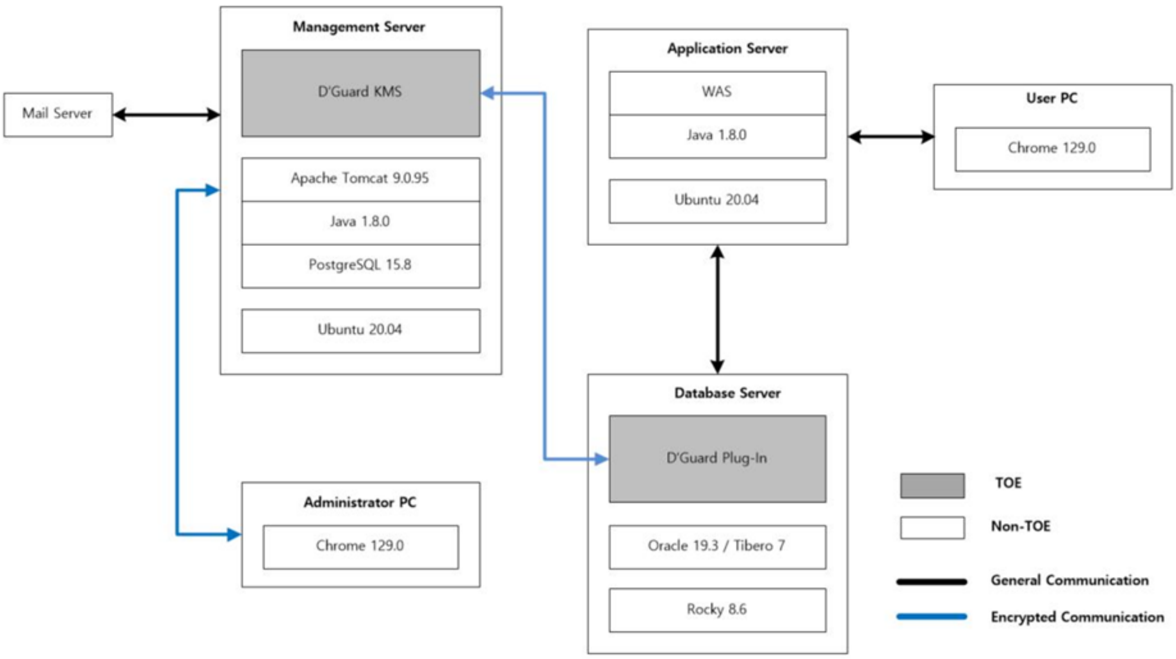
Mail Server

It is used to send information mail to administrator in case of potential security threat of the TOE.

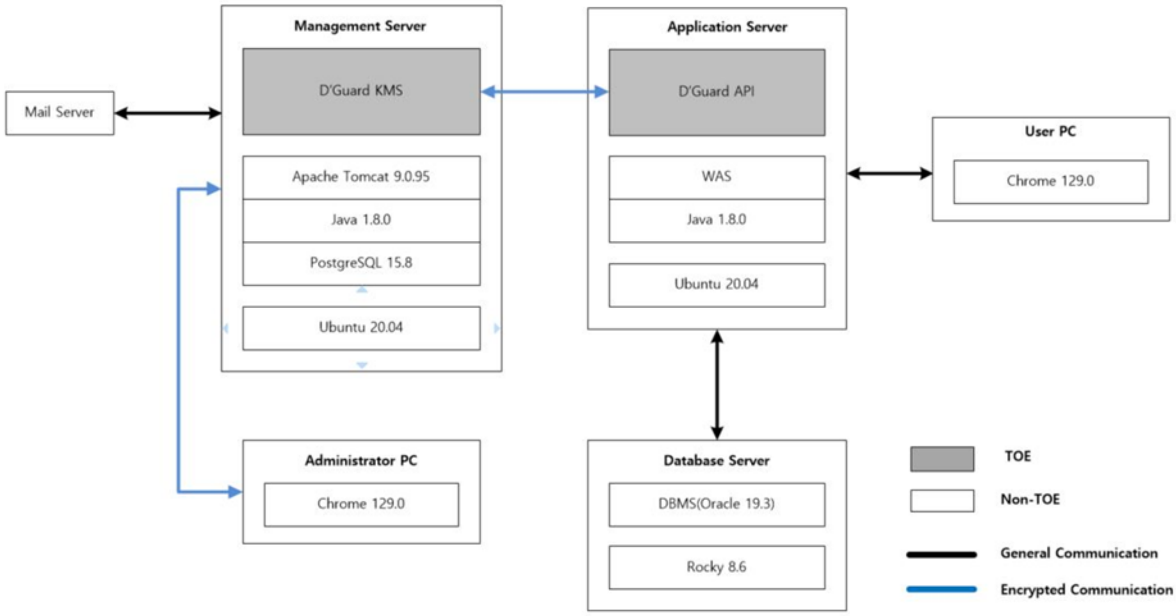
1.4. TOE description

1.4.1. Physical scope of the TOE

The physical scope and boundaries of the TOE are shown in the following figure.



[Figure 3] Physical scope of the TOE (Plug-In type)



[Figure 4] Physical scope of the TOE (API type)

The physical scope of the TOE is divided into the D’Guard KMS, D’Guard Plug-In, D’Guard API and documentation.

[Table 7] Physical composition of the TOE

Composition	Type	Identification information			Within the TOE scope
CD-ROM (1EA)	S/W	TOE Installation Program	D’Guard KMS v5.0.1	D’Guard_KMS_v5.0.1.zip	O
			D’Guard API v5.0.1	D’Guard_API_v5.0.1.zip	O
			D’Guard Plug-In v5.0.1	D’Guard_Plug-In_v5.0.1.zip	O
	Electronic document (PDF)	Guidance document	D’Guard v5.0 KMS Preparatory document and user operation manual v1.2 (D’Guard v5.0 KMS Preparatory document and user operation manual v1.2.pdf)		O
			D’Guard v5.0 Plug-In Preparatory document and user operation manual v1.2 (D’Guard v5.0 Plug-In Preparatory document and user		O

			operation manual v1.2.pdf)	
			D’Guard v5.0 API Preparatory document and user operation manual v1.2 (D’Guard v5.0 API Preparatory document and user operation manual v1.2.pdf)	O

The physical scope of the TOE includes all the information in [Table 7] TOE component information, including software and documentation. The CD includes 3rd party software JRE 8.0, PostgreSQL 15.8, and Apache Tomcat 9.0.95, which are provided for the convenience of the operator. JRE, PostgreSQL, Apache Tomcat are excluded from the scope of the TOE.

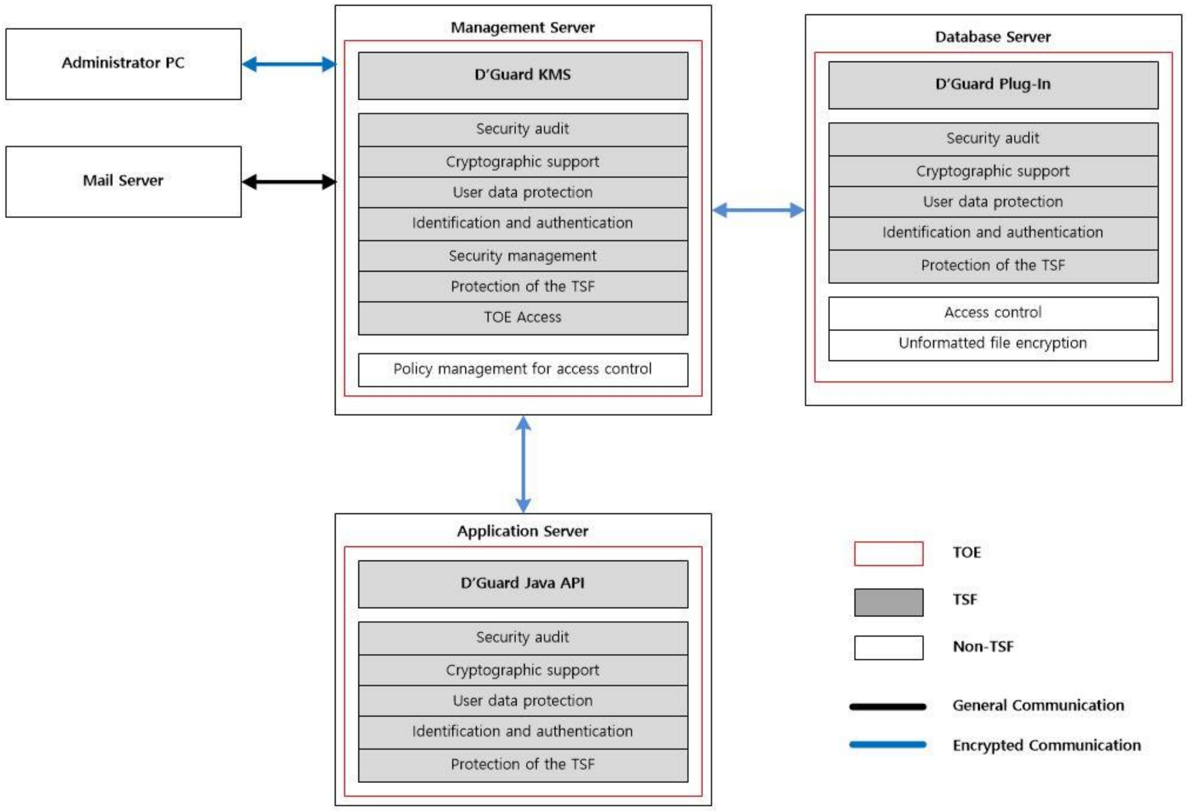
The distributed the TOE package consists of CD 1EA, and CD is provided by direct delivery method.

The validated cryptographic module(KCMVP) used in the TOE is Java module in the D’Guard KMS, the D’Guard API, and C module used in the the D’Guard Plug-In.

[Table 8] The validated cryptographic module(KCMVP) of the TOE

Item	Specification	
Cryptographic module name	MagicCrypto V3.0.0	INISAFE Crypto for C V5.4
Developer	Dreamsecurity Co., Ltd.	INITECH Co., Ltd.
Validation number	CM-200-2026.12	CM-233-2028.6
Module type	S/W	S/W
Validation date	2021. 12. 31	2023. 06. 19
Expiration date	2026. 12. 31	2028. 06. 19
Components	D’Guard KMS D’Guard API	D’Guard Plug-In

1.4.2. Logical scope of the TOE



[Figure 5] Logical scope of the TOE

D’Guard KMS

[Security audit(FAU)]

The security audit function consists of generating audit data, inquiring audit data, analyzing and responding to potential security violations, and protecting audit data. The generation of audit data is generated by including the start and end of the TOE, the result of performing security management for the TOE of the authorized administrator, and audit data collected by the Plug-In agent and API module, and is generated including the date and time of the event, the type of the event, the identity of the subject, and the event result information.

The authorized administrator may selectively check the generated audit data by distinguishing the driving and termination of the TOE and user data encryption according to the optional AND condition, and the generated audit data is protected from unauthorized deletion.

The TOE analyzes potential security violations, such as successive failure of administrator authentication, sends alarm mail to authorized administrators and generates audit data. In addition, by analyzing violations of the threshold of the audit repository, when the primary threshold of 80%

is exceeded, an alarm mail is sent to the authorized administrator, and when the secondary threshold of 90% is exceeded, an alarm mail is sent to the authorized administrator, and after deleting some of the oldest audit data, audit data is generated.

[Cryptographic support(FCS)]

The TOE supports cryptographic key management, cryptographic operation, and random bit generation.

The Encryption key generation for encryption of user data and TSF data is generated using HASH_DRBG_SHA256, which is a random bit generator of the validated cryptographic module.

The generated user data encryption key is encrypted and stored using ARIA256(KCMVP) in the policy DB and integrity verification information is stored together using HMAC-SHA256. For TSF data protection, ARIA256 and HMAC-SHA256 are used to store encryption and integrity verification information.

The management server receives a policy distribution request from the agent, encrypts the security policy including the user data encryption key with the RSAES 2048-bit algorithm of the verified encryption module, and safely distributes it.

TOE destroys the memory in the area in three zeroing procedures according to the encryption key destruction procedure used internally, including the user data encryption key.

[User data protection(FDP)]

The TOE protects against unauthorized attacks by initializing all information used inside the TOE for cryptographic key generation, cryptographic key distribution, cryptographic operation, and random bit generation after use.

In order to protect user data stored in the DBMS to be protected, block encryption algorithms (ARIA-128/256, SEED-128, and LEA-128/256) are encrypted and decrypted according to the security policy set by the administrator authorized through the verified encryption module. In addition, one-way encryption is supported through hash algorithms (SHA-256/512). Once the encryption/decryption is completed, it performs initialization to prevent the restoration of the previous value of the original user data.

It provides a function of encrypting and decrypting user data by column, and prevents the same ciphertext from being generated for the same plaintext when encrypting user data.

[Identification and authentication(FIA)]

The TOE performs the identification and authentication based on user ID and password. All TOE management functions cannot be used before user authentication is performed.

When authenticating a user, password input protects against exposure by displaying only blank or masking characters and does not provide reason for failure when authentication fails. It also protects against authentication reuse attacks by receiving one-time captcha during authentication. In case of continuous authentication failure, time delay is disabled and account lockout is performed according to the set value to protect against unauthorized attack. It also sends alert mails to authorized administrators.

The password combination rule for authentication must be 10 or more and 20 or less digits, and must use three combinations of English characters/number and special characters. The character must not be more than 3 consecutive digits, and cannot be used even if the same character is more than 3 times. Account information should not be included, and the same password as the password used before 3 times cannot be used.

In order to secure the communication interval between TOE components, the management server and the agent exchange each communication encryption key after mutual authentication using the RSA 2048-bit public key pair, and then encrypt and transmit the security policy using ARIA256. Mutual authentication uses a self-implemented method based on public key cryptography.

[Security management(FMT)]

The TOE user is divided into super manager who manages all the security functions of the TOE and security user who can only perform the inquiry function of audit data and security policy. The security management function can be performed by authorized administrator only.

The TOE provides a management function of a console CLI environment and a management function of a web GUI environment. The management function of the console CLI environment is used at the time of initial installation, and after that, it provides encryption and integrity verification data update functions for major information in the configuration file (TSF data). The management function of the web GUI environment provides a security management function including generation of a user data encryption key and inquiry of audit data.

The use of TOE's security management function is limited to authorized administrators who have performed authentication. It also forces the authorized administrator's password to be initialized during the TOE installation process.

[Protection of the TSF(FPT)]

The TOE encrypts using ARIA256, which is a verification algorithm of validated cryptographic module, in order to prevent exposure and modification of data stored in the storage controlled by TSF. The TOE generates and stores integrity verification information using HMAC-SHA256 algorithm. In addition, the TOE encrypts using ARIA256, which is a verification algorithm of validated cryptographic module, to prevent the exposure and modification of transmission data between physically separated TOE components, and performs integrity verification using HMAC-SHA256 algorithm.

The TOE monitors whether the main processes of the TOE operate normally through the TSF's own tests. The TOE performs self-test periodically during startup and operation to send an alarm mail and generate audit data to the authorized administrator when the integrity verification of the configuration file and execution module fails during startup or after normal operation.

[TOE access(FTA)]

The TOE limits the maximum number of simultaneous sessions to one by using the administrative access sessions that attempted access from the terminal designated as accessible IP and limiting the simultaneous access of the same user.

The TOE provides the function to terminate the session when the authorized administrator has not been active for a certain period of time after logging in.

[Additional Function(Non TSF)]

Access control policy management: It provides an access control policy management function by an authorized administrator when performing user data encryption according to date, time, and day of the week based on database account, IP address, and application. This function is a security function of TOE that is excluded from the evaluation range. The access control policy management function based on database account, IP address, and application is not a security requirement required by the national database encryption protection profile (PP).

D'Guard Plug-In

[Security audit(FAU)]

The TOE generates audit data on startup and shutdown, management server interworking policy distribution, TSF self-test, and user data encryption processing results.

The audit data for encryption success and decryption success in the user data encryption processing result is determined whether or not audit data is generated by the setting of an authorized administrator.

[Cryptographic support(FCS)]

The TOE supports the cryptographic key management, cryptographic operation, and random bit generation.

The TOE requests the management server for interlocking distribution and encrypts the security policy including the user data encryption key to receive it safely. In order to distribute the security policy securely, after mutual authentication, the communication encryption key is exchanged. At this time, it is generated by using HASH_DRBG_SHA256, which is a random bit generator of the validated cryptographic module.

The TOE encrypts and stores the security policy including the user data encryption key received from the management server by using ARIA256 and verifies the integrity verification information by using HMAC-SHA256.

[User data protection(FDP)]

The TOE protects against unauthorized attacks by initializing all information used inside the TOE for cryptographic key generation, cryptographic key distribution, cryptographic operation, and random bit generation after use.

The TOE provides an interface for encrypting user data using the user data encryption key distributed from the management server.

[Identification and authentication(FIA)]

To secure the communication between TOE components, the management server and the agent mutually authenticate each other using the public key pair of 2048 bits of RSA. After exchanging the communication encryption key, the security policy is encrypted using ARIA256. The mutual authentication uses a self-implemented method based on public key cryptography.

[Protection of the TSF(FPT)]

The TOE encrypts and stores using ARIA256, which is a verification algorithm of validated cryptographic module, in order to prevent exposure and modification of TSF data used internally for operation. In addition, the TOE encrypts using ARIA256, which is a verification algorithm of validated cryptographic module, to prevent the exposure and modification of transmission data between physically separated TOE components, and performs integrity verification using HMAC-SHA256 algorithm.

The TOE monitors the TOE's major processes for normal operation through the TSF's own tests. The TOE performs self-test periodically during startup and operation to generate audit data when the integrity verification of the configuration file and execution module fails during startup or after normal operation.

D’Guard API

[Security audit(FAU)]

The TOE generates audit data on the management server interworking policy distribution, the TSF self-test results, and user data encryption processing results.

The audit data for encryption success and decryption success in the user data encryption processing result is determined whether or not audit data is generated by the setting of an authorized administrator.

[Cryptographic support(FCS)]

The TOE supports the cryptographic key management, cryptographic operation, and random bit generation.

The TOE receives the interworking distribution request from the management server by encrypting the security policy including the user data encryption key. In order to distribute the security policy securely, after mutual authentication, the communication encryption key is exchanged. At this time, it is generated by using HASH_DRBG_SHA256, which is a random bit generator of the validated cryptographic module.

The TOE encrypts and stores the security policy including the user data encryption key received from the management server by using ARIA256 and verifies the integrity verification information by using HMAC-SHA256.

[User data protection(FDP)]

The TOE protects against unauthorized attacks by initializing all information used inside the TOE for the cryptographic key generation, cryptographic key distribution, cryptographic operation, and random bit generation after use.

The TOE provides an interface for encrypting user data using the user data encryption key distributed from the management server.

[Identification and authentication(FIA)]

To secure the communication between the TOE components, the management server and the agent mutually authenticate each other using the public key pair of 2048 bits of RSA. After exchanging communication encryption key, security policy is transmitted using ARIA256. The mutual authentication uses a self-implemented method based on public key cryptography.

[Protection of the TSF(FPT)]

The TOE encrypts and stores using ARIA256, which is a verification algorithm of the validated cryptographic module, in order to prevent the exposure and modification of the TSF data used internally for operation. In addition, the TOE encrypts using ARIA256, which is a verification algorithm of the validated cryptographic module, to prevent the exposure and modification of transmission data between physically separated TOE components, and performs integrity verification using HMAC-SHA256 algorithm.

The TOE monitors the TOE's major processes for normal operation through the TSF's own tests. The TOE performs self-test periodically during startup and operation to generate audit data when the integrity verification of the configuration file and execution module fails during startup or after normal operation.

1.5. Conventions

The notation, formatting and conventions used in this ST are consistent with the Common Criteria for Information Technology Security Evaluation.

The CC allows several operations to be performed for 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 indicted in square brackets like [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 <u>*underlined and italicized*</u>.

Refinement

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

1.6. Terms and definitions

Terms used in this ST, which are the same as in the CC, must follow those in the CC.

Approved cryptographic algorithm

A cryptographic algorithm selected by Korea Cryptographic Module Validation Authority for block cipher, secure hash algorithm, message authentication code, random bit generation, key agreement, public key cipher, digital signatures cryptographic algorithms considering safety, reliability and interoperability

Application Server

The application server defined in this ST refers to the server that installs and operates the application, which is developed to provide a certain application service by the organization that operates the TOE. The pertinent application reads the user data from the DB, which is located in the database server, by the request of the application service user, or sends the user data to be stored in the DB to the database server.

Approved mode of operation

The mode of cryptographic module using 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 as an attacker’s expertise, resources and motivation

Augmentation

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

Authorized Administrator

Authorized user to securely operate and manage the TOE

Authentication Data

Information used to verify the claimed identity of a user

Authorized User

The TOE user who may, in accordance with the SFRs, perform an operation

Can/could

The ‘can’ or ‘could’ presented in Application notes indicates optional requirements applied to the TOE by ST author’s choice

Column

A set of data values of a particular simple type, one for each row of the table in a relational database

Component

Smallest selectable set of elements on which requirements may be based

Critical Security Parameters (CSP)

Information related to security that can erode the security of the encryption module if exposed or changed (e.g., verification data such as secret key/private key, password, or Personal Identification Number)

Class

Set of CC families that share a common focus

Database

A set of data that is compiled according to a certain structure in order to receive, save, and provide data in response to the demand of multiple users to support multiple application duties at the same time. The database related to encryption by column, which is required by this ST, refers to the relational database.

Database Server

The database server defined in this ST refer to the server in which the DBMS managing the protected DB is installed in the organization that operates the TOE

DBMS (Database Management System)

A software system composed to configure and apply the database. The DBMS related to encryption by column, which is required by this ST, refers to the database management system based on the relational database model

Data Encryption Key (DEK)

Key that encrypts and decrypts the data

Decryption

The act that restoring the ciphertext into the plaintext using the 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

Encryption

The act that converts the plaintext into the ciphertext using the encryption key

Element

Indivisible statement of a security need

External Entity

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

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

Family

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

Identity

Representation uniquely identifying entities (e.g. user, process or disk) within the context of the TOE

Iteration

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

Key Encryption Key (KEK)

Key that encrypts and decrypts another cryptographic key

Management access

The access to the TOE by using the HTTPS, SSH, TLS, etc to manage the TOE by administrator, remotely

Object

Passive entity in the TOE containing or receiving information and on 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

Private Key

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

Public Key (asymmetric) cryptographic algorithm

A cryptographic algorithm that uses a pair of public and private keys

Random bit generator

A device or algorithm that outputs a binary string that is statistically independent and is not biased. The RBG used for cryptographic application generally generates 0 and 1 bit string, and the string can be combined into a random bit block. The RBG is classified into the deterministic and non-deterministic type. The deterministic type RBG is composed of and algorithm that generates bit strings from the initial value called a “seed key,” and the non-deterministic type RBG produces output that depends on the unpredictable physical source.

Recommend/be recommended

The ‘recommend’ or ‘be recommended’ presented in Application notes is not mandatorily recommended, but required to be applied for secure operations of the TOE

Refinement

Addition of details to a component

Role

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

Security Function Policy (SFP)

A Set of rules that describes the specific security action performed by TSF (TOE security functionality) and describe them as SFR (security function requirement)

Secret Key

A cryptographic key which is used in a symmetric cryptographic algorithm and is uniquely associated with one or several entity, not to be disclosed

Security Target (ST)

Implementation-dependent statement of security needs for a specific identified TOE

Security attribute

The characteristics of the subject used to define the SFR, user (including the external IT product), object, information, session and/or resources. These values are used to perform the SFR

Security Token

PBKDF2-based encrypted file for public key pair for mutual authentication and secure communication between the TOE and master key of the TOE

Selection

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

Shall/must

The ‘shall’ or ‘must’ presented in Application notes indicates mandatory requirements applied to the TOE

SSL (Secure Sockets Layer)

This is a security protocol proposed by Netscape to ensure confidentiality, integrity and security over a computer network

Symmetric cryptographic technique

Encryption scheme that uses the same secret key in mode of encryption and decryption, also known as secret key cryptographic technique

Subject

Active entity in the TOE that performs operations on objects

Target of Evaluation (TOE)

Set of software, firmware and/or hardware possibly accompanied by guidance

Threat Agent

Entity that can adversely act on assets

TLS (Transport Layer Security)

This is a cryptographic protocol between a SSL-based server and a client and is described in RFC 2246

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

TSF Data

Data for the operation of the TOE upon which the enforcement of the SFR relies

Unstructured File

File with unstructured form and structure

User

Refer to “External entity”

User Data

Data for the user that does not affect the operation or the TSF

dgfile

A program that encrypts and decrypts all unstructured files

dgsam

This is a program that encrypts and decrypts the value of a specific item by separating the values of each item from the unstructured file by a separator or a fixed length.

1.7. ST organization

This document is structured as below:

Chapter 1 Introduction describes the Security Target and TOE reference, TOE overview, TOE description, conventions and terms and definitions.

Chapter 2 Conformance Claims describes the conformance with the Common Criteria, protection profile and package and presents the conformance rationale and protection profile conformance statement.

Chapter 3 describes the security objectives for the operational environment.

Chapter 4 defines the extended components for the database encryption.

Chapter 5 describes the security functional and assurance requirements.

Chapter 6 describes the TOE summary specification.

2. Conformance claim

2.1. CC conformance claim

[Table 9] CC conformance claim

<table>
<tr><td colspan="2">Common Criteria</td><td>Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5
· Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model, Version 3.1, Revision 5 (CCMB-2017-04-001, April, 2017)
· Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components, Version 3.1, Revision 5 (CCMB-2017-04-002, April, 2017)
· Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components, Version 3.1, Revision 5 (CCMB-2017-04-003, April, 2017)</td></tr>
<tr><td rowspan="3">Conformance claim</td><td>Part 2 Security functional components</td><td>Extended: FCS_RBG.1, FIA_IMA.1, FDP_UDE.1, FMT_PWD.1, FPT_PST.1, FTA_SSL.5</td></tr>
<tr><td>Part 3 Security assurance components</td><td>Conformant</td></tr>
<tr><td>Package</td><td>Augmented: EAL1 augmented (ATE_FUN.1)</td></tr>
</table>

2.2. PP conformance claim

This ST conforms to the "National PP of Database Encryption V1.1"

- ✓ PP Title and Version: National PP for Database Encryption V1.1
- ✓ Certificate No/Data: KECS-PP-0820-2017/2019-12-11
- ✓ Publication Date: 2019-12-11
- ✓ Evaluation Assurance Level: EAL1+(ATE_FUN.1)
- ✓ Conformance Type: Strict PP conformance

2.3. Package conformance claim

This ST claim conformance to assurance package EAL1 augmented with ATE_FUN.1.

2.4. Conformance claim rationale

In this security target specification, the TOE type, security purpose, and security requirements were all the same according to the strict compliance method of 'National Database Encryption Protection Profile V1.1'.

[Table 10] Conformance claim rationale

Standard list	PP	ST	Claim Rationale
TOE Type	Separate into 'Plug-in' and 'API'	Separate into 'Plug-in' and 'API'	Same as PP
Security Purpose	OE.PHYSICAL_CONTROL	OE.PHYSICAL_CONTROL	Same as PP
	OE.TRUSTED_ADMIN	OE.TRUSTED_ADMIN	Same as PP
	OE.SECURE_DEVELOPMENT	OE.SECURE_DEVELOPMENT	Same as PP
	OE.LOG_BACKUP	OE.LOG_BACKUP	Same as PP
	OE.OPERATION_SYSTEM_RE-INFORCEMENT	OE.OPERATION_SYSTEM_RE-INFORCEMENT	Same as PP
	OE.TIME_STAMP	OE.TIME_STAMP	More restrictive than PP PP does not have security issues and security requirements for timestamps used for audit records, but this ST additionally identifies the assumption that a secure

Standard list	PP	ST	Claim Rationale
			timestamp is received from the operational environment of the TOE
	OE.SECURE_DBMS	OE.SECURE_DBMS	More restrictive than PP PP does not have a security problem definition and security requirements for DBMS storing data and audit data, but this ST further identifies the assumption that it operates physically and safely
	OE.TRUSTED_PATH	OE.TRUSTED_PATH	More restrictive than PP PP did not define security issues and requirements for the communication section accessed by the administrator, but this ST further identifies the assumption that information transmitted when the administrator accesses the management server is protected through a secure channel
Security Features Requirement	FAU_ARP.1	FAU_ARP.1	Same as PP
	FAU_GEN.1	FAU_GEN.1	Same as PP
	FAU_SAA.1	FAU_SAA.1	Same as PP
	FAU_SAR.1	FAU_SAR.1	Same as PP

Standard list	PP	ST	Claim Rationale
	FAU_SAR.3	FAU_SAR.3	Same as PP
	FAU_STG.1	-	FAU_STG.3 and FAU_STG.4 have a dependent relationship with FAU_STG.1, but in this ST, audit data is stored in the DBMS that works with TOE, thereby satisfying the dependent relationship in the OE.SECURE_DBMS
	FAU_STG.3	FAU_STG.3	Same as PP
	FAU_STG.4	FAU_STG.4	Same as PP
	FCS_CKM.1(1)	FCS_CKM.1(1)	Same as PP
	FCS_CKM.1(2)	FCS_CKM.1(2)	Same as PP
	FCS_CKM.2	FCS_CKM.2	Same as PP
	FCS_CKM.4	FCS_CKM.4	Same as PP
	FCS_COP.1(1)	FCS_COP.1(1)	Same as PP
	FCS_COP.1(2)	FCS_COP.1(2)	Same as PP
	FCS_RBG.1	FCS_RBG.1	Same as PP
	FDP_UDE.1	FDP_UDE.1	Same as PP
	FDP_RIP.1	FDP_RIP.1	Same as PP
	FIA_AFL.1	FIA_AFL.1	Same as PP
	FIA_IMA.1	FIA_IMA.1	Same as PP
	FIA_SOS.1	FIA_SOS.1	Same as PP
	FIA_UAU.1	FIA_UAU.2	Use hierarchical FIA_UAU.2 in accordance with the

Standard list	PP	ST	Claim Rationale
			application precautions of PP FIA_UAU.1
	FIA_UAU.4	FIA_UAU.4	Same as PP
	FIA_UAU.7	FIA_UAU.7	Same as PP
	FIA_UID.1	FIA_UID.2	Use hierarchical FIA_UID.2 in accordance with the application precautions of PP FIA_UID.1, FIA_UID.2
	FMT_MOF.1	FMT_MOF.1	Same as PP
	FMT_MTD.1	FMT_MTD.1	Same as PP
	FMT_PWD.1	FMT_PWD.1	Same as PP
	FMT_SMF.1	FMT_SMF.1	Same as PP
	FMT_SMR.1	FMT_SMR.1	Same as PP
	FPT_ITT.1	FPT_ITT.1	Same as PP
	FPT_PST.1	FPT_PST.1	Same as PP
	FPT_STM.1	-	Not required in TOE with select SFR
	FPT_TEE.1	-	Not required in TOE with select SFR
	FPT_TST.1	FPT_TST.1	Same as PP
	FTA_MCS.2	FTA_MCS.2	Same as PP
	FTA_SSL.5	FTA_SSL.5	Same as PP
	FTA_TSE.1	FTA_TSE.1	Same as PP
	FTP_ITC.1	-	Not required in TOE with select SFR

Standard list	PP	ST	Claim Rationale
	FTP_TRP.1	-	Not required in TOE with select SFR
Security Purpose	ASE_INT.1	ASE_INT.1	Same as PP
	ASE_CCL.1	ASE_CCL.1	Same as PP
	ASE_OBJ.1	ASE_OBJ.1	Same as PP
	ASE_ECD.1	ASE_ECD.1	Same as PP
	ASE_REQ.1	ASE_REQ.1	Same as PP
	ASE_TSS.1	ASE_TSS.1	Same as PP
	ADV_FSP.1	ADV_FSP.1	Same as PP
	AGD_OPE.1	AGD_OPE.1	Same as PP
	AGD_PRE.1	AGD_PRE.1	Same as PP
	ALC_CMC.1	ALC_CMC.1	Same as PP
	ALC_CMS.1	ALC_CMS.1	Same as PP
	ATE_FUN.1	ATE_FUN.1	Same as PP
	ATE_IND.1	ATE_IND.1	Same as PP
	AVA_VAN.1	AVA_VAN.1	Same as PP

3. Security Objectives

The following are the security objectives handled by technical and procedural method supported from operational environment in order to provide the TOE security functionality accurately.

3.1. Security objectives for the operational environment

OE.PHYSICAL_CONTROL
The place where 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 users, have appropriately trained for the TOE management functions and accurately fulfill the duties in accordance with administrator guidance.

OE.SECURE_DEVELOPMENT
The developer who uses the TOE to interoperate with the user identification and authentication function in the operational environment of the business system shall ensure that the security functions of the TOE are securely applied in accordance with the requirements of the manual provided with the TOE.

OE.LOG_BACKUP
The authorized administrator of the TOE shall periodically checks a spare space of audit data storage in case of the audit data loss, and carries out the audit data backup (external log server or separate storage device, etc.) to prevent audit data loss.

OE.OPERATION_SYSTEM_RE-INFORCEMENT

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

OE.TIME_STAMP

The TOE shall accurately record security-relevant events by using reliable time stamps provided by the TOE operational environment.

OE.SECURE_DBMS

The DBMS that stores TSF data and audit data should be operated physically and safely.

OE.TRUSTED_PATH

Information transmitted when an authorized administrator accesses a web server using a web browser should be protected through a secure channel.

4. Extended components definition

4.1. Cryptographic support (FCS)

4.1.1. Random Bit Generation

Family Behaviour

This family (FCS_RBG, Random Bit Generation) defines requirements for the TSF to provide the capability that generates random bits required for TOE cryptographic operation.

Component Leveling

FCS_RBG.1 random bit generation requires TSF to provide the capability that generates random bits required for TOE cryptographic operation.

Management: FCS_RBG.1

There are no management activities foreseen.

Audit: FCS_RBG.1

There are no auditable events foreseen.

4.1.1.1. FCS_RBG.1 Random bit generation

Hierarchical to	No other components.
Dependencies	No dependencies.
FCS_RBG.1.1	The TSF shall generate random bits required to generate a cryptographic key using the specified random bit generator that meets

the following [assignment: *list of standards*]

4.2. Identification & authentication (FIA)

4.2.1. TOE Internal mutual authentication

Family Behaviour

This family (FIA_IMA, TOE Internal Mutual Authentication) defines requirements for providing mutual authentication between TOE components in the process of user identification and authentication.

Component Leveling

FIA_IMA TOE internal mutual authentication	—	1

FIA_IMA.1 TOE Internal mutual authentication requires that the TSF provides mutual authentication function between TOE components in the process of user identification and authentication.

Management: FIA_IMA.1

There are no management activities foreseen.

Audit: FIA_IMA.1

The following actions are recommended to record if FAU_GEN Security audit data generation family is included in the PP/ST:

a) Minimal: Success and failure of mutual authentication

b) Minimal: 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.1.1	The TSF shall perform mutual authentication between [assignment: *different parts of TOE*] using the [assignment: authentication protocol] that meets the following [assignment: *list of standards*]

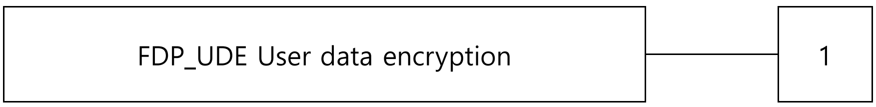
4.3. User data protection (FDP)

4.3.1. User data encryption

Family Behaviour

This family provides requirements to ensure confidentiality of user data.

Component leveling



FDP_UDE.1 User data encryption requires confidentiality of user data.

Management: FDP_UDE.1

The following actions could be considered for the management functions in FMT:

a) Management of user data encryption/decryption rules

Audit: FDP_UDE.1

The following actions are recommended to record if FAU_GEN Security audit data generation is included in the PP/ST:

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

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 TOE users with the ability to encrypt/decrypt user data according to [assignment: *the list of encryption/decryption method*] specified.

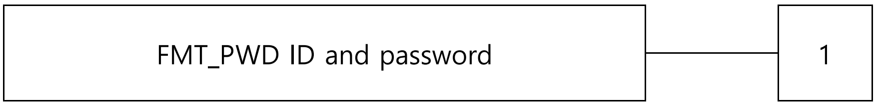
4.4. Security Management (FMT)

4.4.1. ID and password

Family Behaviour

This family defines the capability that is required to control ID and password management used in the TOE, and set or modify ID and/or password by authorized users.

Component leveling



FMT_PWD.1 ID and password management, requires that the TSF provides the management function of ID and password.

Management: FMT_PWD.1

The following actions could be considered for the management functions in FMT:

a) Management of ID and password configuration rules.

Audit: FMT_PWD.1

The following actions are recommended to record if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: All changes of the password.

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 the ability to manage the password of [assignment: *list of functions*] to [assignment: *the authorized identified roles*]. 1. [assignment: *password combination rules and/or length*] 2. [assignment: *other management such as management of special characters unusable of password, etc.*]
FMT_PWD.1.2	The TSF shall restrict the ability to manage the ID of [assignment*: list of functions*] to [assignment: *the authorized identified roles*]. 1. [assignment: *ID combination rules and/or length*] 2. [assignment: *other management such as management of special characters unusable for ID, etc.*]
FMT_PWD.1.3	The TSF shall provide the capability for [selection, choose one of: *setting ID and password when installing, setting password when installing, changing the ID and password when the authorized administrator access for the first time, changing the password when the authorized administrator access of the first time*].

4.5. Protection of the TSF (FPT)

4.5.1. Protection of stored TSF data

Family Behaviour

This family defines rules to protect TSF data stored within containers controlled by the TSF from the unauthorized modification or disclosure.

Component leveling

FPT_PST Protection of stored TSF data — 1

FPT_PST.1 Basic protection of stored TSF data, requires the protection of TSF data stored in containers controlled by the TSF.

Management: FPT_PST.1

There are no management activities foreseen.

Audit: FPT_PST.1

There are no auditable events foreseen.

4.5.1.1. FPT_PST.1 Basic protection of stored TSF data

Hierarchical to	No other components.
Dependencies	No dependencies.
FPT_PST.1.1	The TSF shall protect [assignment: *TSF data*] stored in containers controlled by the TSF from the unauthorized [selection: *disclosure, modification*].

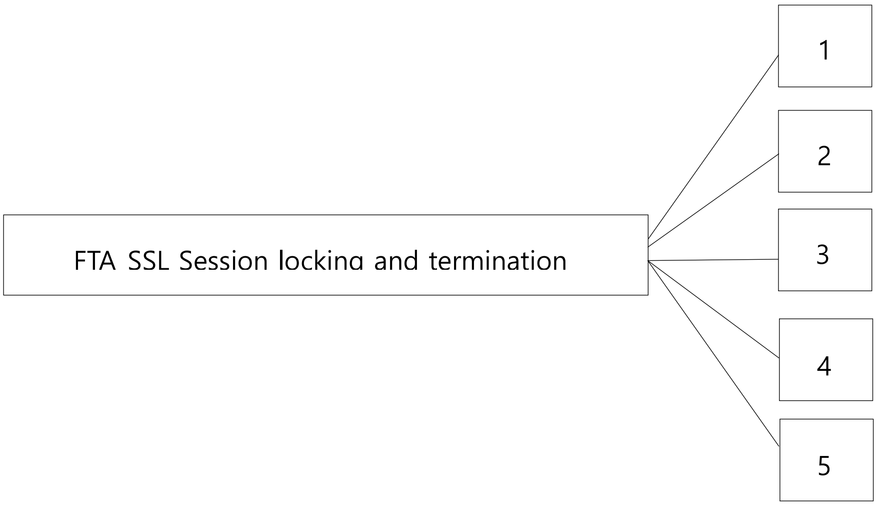
4.6. TOE Access (FTA)

4.6.1. Session locking and termination

Family Behaviour

This family (FTA_SSL, Session locking and termination) defines requirements for the TSF to provide the capability for TSF-initiated and user-initiated locking, unlocking, and termination of interactive sessions.

Component leveling



In CC Part 2, the session locking and termination family consists of four components. In this ST, it consists of five components by extending one additional component as follows.

※ The relevant description for four components contained in CC Part 2 is omitted.

FTA_SSL.5 The management of TSF-initiated sessions, provides requirements that the TSF locks or terminates the session after a specified time interval of user inactivity.

Management: FTA_SSL.5

The following actions could be considered for the management functions in FMT:

a) Specification for the time interval of user inactivity that is occurred the session locking and termination for each user

b) Specification for the time interval of default user inactivity that is occurred the session locking and termination

Audit: FTA_SSL.5

The following actions are recommended to record if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Locking or termination of interactive session

4.6.1.1. FTA_SSL.5 Management of TSF-initiated sessions

Hierarchical to	No other components.
Dependencies	[FIA_UAU.1 authentication or No dependencies.]
FTA_SSL.5.1	The TSF shall [selection: · *lock the session and re-authenticate the user before unlocking the session* · *terminate*] an interactive session after a [assignment: *time interval of user inactivity*]

5. Security requirements

The security requirements specify security functional requirements and assurance requirements that must be satisfied by the TOE that claims conformance to this ST.

The security functional requirements included in this ST are derived from CC Part 2 and Chapter 4 Extended Components Definition.

The following table summarizes the security functional requirements used in the ST

[Table 6] Security functional requirements

Security functional class	Security functional component	
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.3	Action in case of possible audit data loss
	FAU_STG.4	Prevention of audit data loss
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(Extended)	Random bit generation

FDP	FDP_UDE.1(Extended)	User data encryption
	FDP_RIP.1	Subset residual information protection
FIA	FIA_AFL.1	Authentication failure handling
	FIA_IMA.1(Extended)	TOE Internal mutual authentication
	FIA_SOS.1	Verification of secrets
	FIA_UAU.2	Timing of authentication
	FIA_UAU.4	Single-use authentication mechanisms
	FIA_UAU.7	Protected authentication feedback
	FIA_UID.2	User authentication prior to all actions
FMT	FMT_MOF.1	Management of security functions behavior
	FMT_MTD.1	Management of TSF data
	FMT_PWD.1(Extended)	Management of ID and password
	FMT_SMF.1	Specification of management functions
	FMT_SMR.1	Security roles
FPT	FPT_ITT.1	Basic internal TSF data transfer protection
	FPT_PST.1(Extended)	Basic protection of stored TSF data
	FPT_TST.1	TSF testing
FTA	FTA_MCS.2	Per user attribute limitation on multiple concurrent sessions
	FTA_SSL.5(Extended)	Management of TSF-initiated sessions
	FTA_TSE.1	TOE session establishment

5.1. Security functional requirements (Mandatory SFRs)

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 [*sending a warning email to the authorized administrator*] upon detection of a potential security violation.

5.1.1.2. FAU_GEN.1 Audit data generation

Hierarchical to	No other components.
Dependencies	FPT_STM.1 Reliable time stamps
FAU_GEN.1.1	The TSF shall be able to generate an audit record of the following auditable events: a) Start-up and shutdown of the audit functions b) All auditable events for the *not specified* level of audit; and c) [Refer to the "auditable events" in [Table 12] Audit events, *no other components*].
FAU_GEN.1.2	The TSF shall record within each audit record at least the following information: a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST [Refer to the contents of "additional audit record" in [Table 12] Audit events, *no other components*].

[Table 7] Audit events

Security functional class	Auditable event	Additional audit record
FAU_ARP.1	Actions taken due to potential security violations	
FAU_SAA.1	Enabling and disabling of any of the analysis mechanisms, Automated responses performed by the tool	
FAU_STG.3	Actions taken due to exceeding of a threshold	

FAU_STG.4	Actions taken due to the audit storage failure	
FCS_CKM.1(1)	Success and failure of the activity	
FCS_CKM.2	Success and failure of the activity (only applying to distribution of key related to user data encryption/decryption)	
FCS_CKM.4	Success and failure of the activity (only applying to destruction of key related to user data encryption/decryption)	
FCS_COP.1(1)	Success and failure of the activity	
FDP_UDE.1	Success and failure of user data encryption/decryption	
FIA_AFL.1	The reaching of the threshold for the unsuccessful authentication attempts and the actions taken, and the subsequent, if appropriate, restoration to the normal state	
FIA_IMA.1	Success and failure of mutual authentication Modify of authentication protocol	
FIA_UAU.2	All use of the authentication mechanism	
FIA_UAU.4	Attempts to reuse authentication data	
FIA_UID.2	All use of the user identification mechanism, Including the user identity provided	
FMT_MOF.1	All modifications in the behavior of the functions in the TSF	
FMT_MTD.1	All modifications to the values of TSF data	Modified values of TSF data
FMT_PWD.1	All changes of the password	
FMT_SMF.1	Use of the management functions	
FMT_SMR.1	Modifications to the user group of rules divided	
FPT_TST.1	Execution of the TSF self-tests and the results of the tests	Modified TSF data or execution code in case of integrity violation
FTA_MCS.2	Denial of a new session based on the limitation of multiple concurrent sessions	

FTA_SSL.5	Locking or termination of 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 in monitoring the audited events and based upon these rules indicate a potential violation of the enforcement of the SFRs.

FAU_SAA.1.2 The TSF shall enforce the following rules for monitoring audited events:

a) [Potential violation analysis
   - Among the audit cases of FIA_UAU.2, the audit case of failure of certification, the audit case of integrity violation among the audit cases of FPT_TST.1, and the failure of self-test of the verified cryptographic module(KCMVP)
   - Audit evidence storage threshold exceeded event in FAU_STG.3 and audit evidence storage saturation event in FAU_STG.4
   ]
b) [Other audit event rules including potential violations
   - FIA_UAU.2: duplicate authentication failures occurred during an administrator's authentication failure audit (Account deactivation: 5-minute authentication delay when authentication fails more than 5 times)
   - FIA_STG.3: limit of the audit trail exceeds 80%
   - FIA_STG.4: limit of the audit trail saturates 90%
   - Self-test failure incident of verified cryptographic module(KCMVP)
   ]

5.1.1.4. FAU_SAR.1 Audit review

Hierarchical to No other components.

Dependencies FAU_GEN.1 Audit data generation

FAU_SAA.1.1 The TSF shall provide [authorized administrator] with the capability to read [all the audit data] from the audit records.

FAU_SAA.1.2 The TSF shall provide the audit records in a manner suitable for the authorized administrator to interpret the information.

5.1.1.5. 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 capability to apply [*methods of selection and/or ordering*] of audit data based on [*criteria with logical relations*]

[

- Criteria with logical relations: Combination with AND operation when entering the selected criteria events in *[Table 13] Auditable events and selection criteria*
- Methods of selection and/or ordering: Select audit data according to *[Table 13] Auditable events and selection criteria* and order audit data in descending order based on time of occurrence

]

[Table 8] Auditable events and selection criteria

Selection Criteria	Operation	Remarks
Incident date	Select (Start ~ End)	· Criteria of audit data creation date (day, month, year)
IP	Search (IP address)	· D'Guard KMS's IP · D'Guard KMS GUI's connect administrator IP · D'Guard KMS CLI's connect administrator IP · D'Guard Plug-In's IP · D'Guard API's IP
User account	Search (Identifiable user account)	· D'Guard KMS's connect administrator account · D'Guard KMS GUI's Connect administrator account · D'Guard Plug-In's security token account · D'Guard API's Security token account

Result	Select (all/true/fail)	· Processing result of audit data action

5.1.1.6. FAU_STG.3 Action in case of possible audit data loss

Hierarchical to	No other components.
Dependencies	FAU_STG.1 Protected audit trail storage
FAU_STG.3.1	The TSF should take [*Notify the authorized administrator, [None]*] if the audit trail exceeds that when reached threshold [80%] of audit storage.

5.1.1.7. 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 Protected audit trail storage
FAU_STG.4.1	The TSF should *overwrite the oldest stored audit records* and execute [*Send mail to an authorized administrator*] if the audit trail is full.

Application notes

- ✓ When the audit storage saturation setting is exceeded (90%), the oldest audit record is deleted with the number of audit data corresponding to 5% of the total count of audit records, and the audit data is generated and an alert email is sent to the authorized administrator.

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(1) Cryptographic operation]** FCS_CKM.4 Cryptographic key destruction **FCS_RBG.1 Random bit generation**

FCS_CKM.1.1	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [*"Cryptographic algorithm" of [Table 14] User data encryption algorithm and key length*] and specified cryptographic key sizes [*"Cryptographic key sizes" of [Table 14] User data encryption algorithm and key length*] that meet the following: [*"Standard list" of [Table 15] Encryption key generation*].

[Table 9] Encryption key generation

Standard list	Cryptographic operations	Cryptographic algorithm	Complexity of cryptographic	Cryptographic key sizes	Purpose
ISO/IEC 18031	Random bit generator	HASH-DRBG-SHA256	256	-	For generating user data and TSF data encryption keys

[Table 15] User data encryption algorithm and key length

Standard list	Cryptographic operations	Cryptographic algorithm	Complexity of cryptographic	Cryptographic key sizes	Purpose
KS X 1213-1	Symmetric key encryption	ARIA (CBC)	128	128	Encrypt user data (symmetric key ciphering)
			256	256	
TTAS.KO-12.0004/R1	Symmetric key encryption	SEED (CBC)	128	128	
TTAK.KO-12.0223	Symmetric key encryption	LEA(CBC)	128	128	
			256	256	
ISO/IEC 10118-3	Secure Hash	SHA-2	256	None	Encrypt user data (HASH)
			512	None	
TTAK.KO-12.0191	Random bit generator	HASH-DRBG-SHA256	256	-	Random IV (Use to prevent the same ciphertext from being generated in the

					same plaintext)

Application notes

- ✓ The key length generated by the Random bit generator (HASH-DRBG-SHA256) is 32 bytes with a default value of 8 (quantitative number of 8). Entering the length according to the cipher ratio as a factor produces a key of that length.

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(2) Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction
FCS_RBG.1 Random bit generation

FCS_CKM.1.1 — The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [*"Cryptographic algorithm" of [Table 16] TSF data cryptographic key algorithm and key sizes*] and specified cryptographic key sizes [*"Cryptographic key sizes" of [Table 16] TSF data cryptographic key algorithm and key sizes*] that meet the following: [*"Standard list" of [Table 15] Encryption key generation*].

[Table 16] TSF data cryptographic key algorithm and key sizes

TOE Module	Standard list	Cryptographic algorithm	Cryptographic key sizes	Cryptographic operations	Describe the key type and purpose	Cryptographic key creation
D'Guard KMS	KS X 1213-1	ARIA (CBC)	256	Symmetric key encryption	Used to encrypt KMS token files with KEK	PBKDF2(PIN)
	KS X 1213-1	ARIA (CBC)	256	Symmetric key encryption	Used to store Agent token files encryption with KEK	
	ISO/IEC	SHA-2	None	Secure	Use PBKDF2	-

	10118-3	(256)		Hash		
	KS X 1213-1	ARIA (CBC)	256	Symmetric key encryption	Used to encrypt security policies, audit log and settings file with master key	Using the random bit generator
	ISO/IEC 10118-3	SHA-2 (256)	None	Secure Hash	Used administrator password one-way encryption	-
	KS X 1213-1	ARIA (CBC)	256	Symmetric key encryption	For encrypting user data cryptographic keys in memory with master key	Using the random bit generator
	ISO/IEC 18033-2	RSAES	2048	Public key encryption	For mutual authentication and session key exchange	Using the RSA key generation algorithm
	KS X 1213-1	ARIA (CBC)	256	Symmetric key encryption	Used session key for interval encrypt between TOE	Using the random bit generator
	ISO/IEC 9797-2	HMAC (SHA-2)	256	Integrity Verification	Configuration files, communication built data integrity verification	-
Agent (D’Guard Plug-In)	KS X 1213-1	ARIA (CBC)	256	Symmetric key encryption	Used to decrypt Agent token files with file KEK	PBKDF2(PIN)
	ISO/IEC 10118-3	SHA-2 (256)	None	Secure Hash	Use PBKDF2	-
	KS X 1213-1	ARIA (CBC)	256	Symmetric key encryption	Encrypt key information such as configuration files with agent token keys	Using the random bit generator
	ISO/IEC 18033-2	RSAES	2048	Public key encryption	For mutual authentication and session key exchange	Using the RSA key generation algorithm
	KS X	ARIA	256	Symmetric	Used session key for	Using the

	1213-1	(CBC)		key encryption	interval encrypt between TOE	random bit generator
	ISO/IEC 9797-2	HMAC (SHA-2)	256	Integrity Verification	Configuration files, communication built data integrity verification	-
Agent (D'Guard API)	KS X 1213-1	ARIA (CBC)	256	Symmetric key encryption	Used to decrypt Agent token files with file KEK	PBKDF2(PIN)
	ISO/IEC 10118-3	SHA-2 (256)	None	Secure Hash	Use PBKDF2	-
	KS X 1213-1	ARIA (CBC)	256	Symmetric key encryption	Encrypt key information such as configuration files with agent token keys	Using the random bit generator
	ISO/IEC 18033-2	RSAES	2048	Public key encryption	For mutual authentication and session key exchange	Using the RSA key generation algorithm
	KS X 1213-1	ARIA (CBC)	256	Symmetric key encryption	Used session key for interval encrypt between TOE	Using the random bit generator
	ISO/IEC 9797-2	HMAC (SHA-2)	256	Integrity Verification	Configuration files, communication built data integrity verification	-

Application notes

- ✓ Master Key: It is a key used by D'Guard KMS to encrypt security policies and encryption keys.
- ✓ Agent Token Key: Key that is encrypted in the agent security token and is used to encrypt key information, including security policies and encryption keys

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 attribute, or
FDP_ITC.2 Import of user data with security attribute, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method [*"Distribution method" of [Table 17] Cryptographic key distribution method*] that meets the following: [*"Standard list" of [Table 17] Cryptographic key distribution method*].

[Table 17] Cryptographic key distribution method

Standard list	Cryptographic algorithm	Cryptographic key sizes	Distribution method
ISO/IEC 18031	HASH-DRBG-SHA256	256	Generate communication cryptographic key
ISO/IEC 18033-2	RSAES	2048	Encrypt communication cryptographic key used for communication section encryption. And mutual authentication.
KS X 1213-1	ARIA	256	Encrypt security policy including user data cryptographic key.
ISO/IEC 9797-2	HMAC-SHA256	256	TSF data transmission and integrity check.

5.1.2.4. FCS_CKM.4 Cryptographic destruction

Hierarchical to No other components

Dependencies [FDP_ITC.1 Import of user data without security attribute, or
FDP_ITC.2 Import of user data with security attribute, or
FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4.1 The TSF shall destruct cryptographic keys in accordance with a specified cryptographic key destruction method [*"Destruction method" of [Table 18] [Table 19] Cryptographic key destruction method*] that meets the following: [*"Standard list" of [Table 18] [Table 19]*

Cryptographic key destruction method].

[Table 18] Cryptographic key destruction method of D’Guard KMS

Standard list	Destruction object	Cryptographic key storage location	Destruction method	Detailed method of destruction	Destruction point
None	User data cryptographic key	DB	deletion	Execute SQL to delete form DB	When the master administrator deletes the cryptographic key
None	Session key	Memory	Memory zeroization	Overwrite key memory area 3 times in a row with 0x00	When calling communication shut-down
None	Master key	Memory	Memory zeroization	Overwrite key memory area 3 times in a row with 0x00	When calling process shut-down
None	Security policy information	Memory	Memory zeroization	Overwrite key memory area 3 times in a row with 0x00	After policy deployment communication completes

[Table 19] Cryptographic key destruction method of Agent(Plug-In Agent, API Module)

Standard list	Destruction object	Cryptographic key storage location	Destruction method	Detailed method of destruction	Destruction point
None	User data cryptographic key	Memory	Memory zeroization	Overwrite key memory area 3 times	After the cryptographic operation is

				in a row with 0x00	complete
None	Program KEK	Memory	Memory zeroization	Overwrite key memory area 3 times in a row with 0x00	When calling process shut-down
None	Program DEK	Memory	Memory zeroization	Overwrite key memory area 3 times in a row with 0x00	When calling process shut-down
None	Session key	Memory	Memory zeroization	Overwrite key memory area 3 times in a row with 0x00	When calling communication shut-down

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 attribute, or
FDP_ITC.2 Import of user data with security attribute, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1 The TSF shall perform [*"list of cryptographic operations" of [Table 15] User data encryption algorithm and key length*] in accordance with a specified cryptographic algorithm [*"Cryptographic algorithm" of [Table 15] User data encryption algorithm and key length*] and cryptographic key sizes [*"Cryptographic key sizes" of [Table 15] User data encryption algorithm and key length*] that meet the following: [*"Standard list" of [Table 15] User data encryption algorithm and key length].*

5.1.2.6. FCS_COP.1(2) Cryptographic operation (TSF data encryption)

Hierarchical to No other components

Dependencies	[FDP_ITC.1 Import of user data without security attribute, or FDP_ITC.2 Import of user data with security attribute, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
FCS_COP.1.1	The TSF shall perform [*"Cryptographic operations" of [Table 16] TSF data cryptographic key algorithm and key sizes*] in accordance with a specified cryptographic algorithm [*"Cryptographic algorithm" of [Table 16] TSF data cryptographic key algorithm and key sizes*] and cryptographic key sizes [*"Cryptographic key sizes" of [Table 16] TSF data cryptographic key algorithm and key sizes*] that meet the following: [*"Standard list" of [Table 16] TSF data cryptographic key algorithm and key sizes*].

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 random bits required to generate an cryptographic key using the specified random bit generator that meets the following [[*Table 20] Random bit generator "Standard list"*]

[Table 20] Random bit generator

Standard list	Random bit generator	Base function
ISO/IEC 18031	HASH-DRBG-SHA256	HASH function

5.1.3. User data protection (FDP)

5.1.3.1. FDP_UDE.1 User data encryption (Extended)

Hierarchical to	No other components
Dependencies	FCS_COP.1 Cryptographic operation

FCS_UDE.1.1	The TSF shall provide a function that can encrypt/decrypt the user data to the TOE user according to the specified [encryption/decryption method by column, *[None]*].

5.1.3.2. FDP_RIP.1 Subset residual information protection

Hierarchical to	No other components
Dependencies	No dependencies
FDP_RIP.1.1	The TSF shall ensure that any previous information content of a resource is made unavailable upon the *allocation of the resource to, deallocation of the resource from* the following objects: [user data].

5.1.4. Identification and authentication (FIA)

5.1.4.1. FIA_AFL.1 Authentication failure handling

Hierarchical to	No other components
Dependencies	FIA_UAU.1 Timing of authentication
FIA_AFL.1.1	The TSF shall detect when [5] unsuccessful authentication attempts occur related to [*Administrator authentication attempts*].
FIA_AFL.1.2	When the defined number of unsuccessful authentication attempts has been *met*, the TSF shall [list of actions] [· *Disable identification and authentication (fixed value of 5 minutes)* · *Send alert mail to administrator*]

Application notes

- ✓ If the administrator's authentication attempt occurs more than 5 times, the identification and authentication functions are disabled, so normal login can be performed after 5 minutes.

5.1.4.2. FIA_IMA.1 TOE Internal mutual authentication (Extended)

Hierarchical to	No other components.
Dependencies	No dependencies.
FIA_IMA.1.1	The TSF shall perform mutual authentication using [*The internally implemented authentication protocol (RSA-2048bit) using the validated cryptographic module*] in accordance with [*None*] between [*[Table21] TOE internal mutual authentication “Mutual verification interval among the TOE components”*].

[Table 21] TOE internal mutual authentication

Mutual verification interval among the TOE components		Validated cryptographic module
D’Guard KMS	D’Guard Plug-In	Refer to *[Table 8] The validated cryptographic module of the TOE*
D’Guard KMS	D’Guard API	

Application notes

- ✓ The TOE performs mutual authentication between its physically separated TOE components regardless of the operation method through its internally implemented authentication protocol. The internally implemented authentication protocol performs mutual authentication with RSA (2048 bit) public key encryption and exchanges session keys.
  1) Generating a security token (including public key pair) for the agent in D'Guard KMS
  2) Manual distribution of the agent's public key pair to the agent
  3) Encrypt the random bit in the agent with D'Guard KMS public key and send it
  4) Decrypt and confirm with own secret key in D'Guard KMS
  5) Encrypt the random bit in the D’Guard KMS with the agent public key and send it
  6) Decrypt and confirm with own secret key in the agent
  7) D'Guard KMS and the agent combine the two random bit exchanged with each other, and then exchange and complete the session key.

5.1.4.3. 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 verify that secrets meet [*The following defined acceptance criteria*].

[

a) Minimum/Maximum length
   - 10 to 20 digits

b) Combination rules
   - Combination of 3 or more among English alphabets/ numbers / special characters
   - Excluded if a password has 3 or more successive letters (abc, def, 123, 432, etc.)
   - Excluded if a password has 3 or more successive letters on the keyboard (asd, qwe, jkl, etc.)
   - Excluded if a password has 3 or more same letters (aaa, 222, 999, etc.)
   - Excluded if a password contains administrator account information
   - Excluded if a password was used the last three times (3 times by default)

c) Change interval (the period during which the password is used)
   - 60 days by default, the interval is defined by the authorized administrator

]

5.1.4.4. FIA_UAU.2 User authentication prior to all actions

Hierarchical to FIA_UAU.1 Timing of authentication

Dependencies FIA_UID.1 Timing of identification

FIA_UAU.2.1 The TSF shall successfully authenticate the authorized administrator before allowing any other TSF-mediated actions on behalf of the authorized administrator.

5.1.4.5. FIA_UAU.4 Single-use authentication mechanisms

Hierarchical to No other components.

Dependencies	No dependencies.
FIA_UAU.4.1	The TSF shall prevent reuse of authentication data related to [*The following identified authentication mechanisms*]. [When the administrator of the TOE authenticates, the captcha, which is one-time verification data, is generated for each session to prevent reuse of the authentication data. The default captcha is valid for 120 seconds. · The console captcha: Numbers 5 digits · The web captcha: English alphabets (small letters)/ numbers 5 digits]

5.1.4.6. FIA_UAU.7 Protected authentication feedback

Hierarchical to	No other components.
Dependencies	FIA_UAU.2 User authentication prior to all actions
FIA_UAU.7.1	The TSF shall provide only [*‘*’ characters, no input characters displayed*] to the user while the authentication is in progress.

Application notes

- ✓ The web authentication: The replacement character ‘*’ is displayed on the password input screen, and ‘The Login information does not match’ is displayed on the screen when authentication fails.
- ✓ The console authentication: The Password input character is not displayed, and if authentication fails, 'Login Failed' is displayed on the screen.

5.1.4.7. FIA_UID.2 User identification prior to all actions

Hierarchical to	FIA.UID.1 Timing of identification
Dependencies	No dependencies.

FIA_UID.2.1	The TSF successfully identifies each authorized administrator before allowing all other actions mediated by the TSF on behalf of the authorized administrator.

5.1.5. Security management (FMT)

[Table 22] Security management action and management type by component

Security functional component	Management function	Management type
FAU_ARP.1	Management of actions (addition, removal, modification)	Management of security functions
FAU_SAA.1	Maintenance of the rules (addition, deletion and modification of the rules in the rule group)	Management of security functions
FAU_SAR.1	Maintenance (deletion, modification, addition) of the group of users with read access right to the audit records	Management of security roles
FAU_STG.3	Maintenance of threshold	Management of TSF data threshold
	Maintenance (deletion, modification, addition) of actions to be taken in case of imminent audit storage failure	Management of security functions
FAU_STG.4	Maintenance (deletion, modification, addition) of actions to be taken in case of audit storage failure	Management of security functions
FDP_UDE.1	Management of the user data encryption/decryption rules	Management of security functions
FIA_AFL.1	Management of the threshold for unsuccessful authentication attempts	Management of TSF data threshold
	Management of actions to be taken in the event of an authentication failure	Management of security functions
FIA_IMA.1	Management of the authentication protocol for mutual authentication	Management of security functions
FIA_SOS.1	Management of the metric used to verify the secrets	Management of security functions
FIA_UAU.2	Management of the authentication data by an administrator, Management of the authentication data by the associated user	Management of TSF data

	Management of the list of actions that can be taken before the user is authenticated	Management of security functions
FIA_UID.2	Management of the user identities	Management of TSF data
	If an authorized administrator can change the actions allowed before identification, the managing of the action lists	Management of security functions
FMT_MOF.1	Management of the group of roles that can interact with the functions in the TSF	Management of security roles
FMT_MTD.1	Management of the group of roles that can interact with the TSF data	Management of security roles
FMT_PWD.1	Management of ID and password configuration rules	Management of security functions
FMT_SMR.1	Management of the group of users that are part of a role	Management of security roles
FMT_ITT.1	Management of the types of modification against which the TSF should protect Management of the mechanism used to provide the protection of the data in transit between different parts of the TSF	Management of security functions
FPT_TST.1	Management of the conditions under which TSF self-testing occurs, such as during initial start-up, regular interval, or under specified conditions Management of the time interval if appropriate	Management of TSF data
FTA_MCS.2	Management of the maximum allowed number of concurrent user sessions by an administrator	Management of TSF data threshold
FTA_SSL.5	Specification of the time of user inactivity after which lock-out occurs for an individual user Specification of the default time of user inactivity after which lock-out occurs	Management of TSF data
FTA_TSE.1	Management of the session establishment conditions by the authorized administrator	Management of TSF data

5.1.5.1. FMT_MOF.1 Management of security functions behavior

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 ability to <u>*conduct management actions of*</u> the functions [*[Table 23] List of security functions behavior of administrator*] to [*the authorized administrators*]

[Table 23] List of security functions behavior of administrator

Administrator Type	Classification	Security Function	Ability			
			Determine the behavior	Disable	Enable	Modify the behavior
Security user	Resource management	Common DB management	○	-	-	-
		Security token management	○	-	-	-
	Policy management	Encryption policy management	○	-	-	-
Super manager	Resource management	Common DB management	○	-	-	○
		Security token management	○	-	-	○
		Configuration management	○	-	-	○
	Policy management	Encryption policy management	○	○	○	○
		Security policy distribution	○	-	-	-
Security user Super manager	Additional management	Cryptographic operation status	○	-	-	-
		Audit data review	○	-	-	-
Super manager (Console)	Initialization	TOE initialization setting	○	-	○	-
	Integrity validation	Integrity validation file modification	○	-	○	-

5.1.5.2. FMT_MTD.1 Management of TSF data

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 ability to _manage_ [*the following list of TSF data*] to [*the authorized administrators*]

[

The TSF data management of *[Table 24] List of TSF data and management ability* is restricted to authorized administrators.

]

[Table 24] List of TSF data and management ability

Administrator Type	TSF data	Ability				
		query	modify	delete	generate	initialize
Security user	Account information	O	-	-	-	-
	Encryption policy	O	-	-	-	-
	Audit data	O	-	-	-	-
	Set value	-	-	-	-	-
Super manager (Web GUI)	Account information	O	O	O	O	-
	Encryption policy	O	O	O	O	-
	Audit data	O	-	-	-	-
	Set value	O	O	O	O	-
Super manager (Console CLI)	Account information	-	-	-	-	O
	Encryption policy	-	-	-	-	-
	Audit data	-	-	-	-	-
	Set value	-	O	-	-	O

Application notes

✓ The super manager must initialize the security policy and audit data to start and operate D’Guard KMS in the console CLI environment, and generate an encryption key such as a master key.

5.1.5.3. FMT_PWD.1 Management of ID and password (Extended)

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 ability to manage the password of [*Administrator of the TOE*] to [*the authorized administrator*]. 1. [None] 2. [None]
FMT_PWD.1.2	The TSF shall restrict the ability to manage the ID of [*Administrator of the TOE*] to [*the authorized administrator*]. 1. [None] 2. [None]
FMT_PWD.1.3	The TSF shall provide the capability for [*changing the password when the authorized administrator accesses for the first time*]

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 capable of performing the following management functions: [*list of management functions to be provided by the TSF*] [· Security function management: Management functions as specified in FMT_MOF.1 · TSF data management: Management functions as specified in FMT_MTD.1 · Password management: Management functions as specified in FMT_PWD.1]

5.1.5.5. FMT_SMR.1 Security roles

Hierarchical to	No other components.

Dependencies	FIA_UID.2 User identification prior to all actions

FMT_SMF.1.1 The TSF shall maintain the roles [*the following authorized roles*].

[

- Super manager: Performs all management functions such as generating and deleting a security user account with the top-level administrative account (both web and console environments)
- Security user: It does not perform the change function, but only the query function. Performing audit query and monitoring functions (web environment only)

]

FMT_SMR.1.2 The TSF shall be able to associate users and their roles defined in FMT_SMR.1.

5.1.6. Protection of the TSF

5.1.6.1. FPT_ITT.1 Basic internal TSF data transfer protection

Hierarchical to	No other components.
Dependencies	No dependencies.

FPT_ITT.1.1 The TSF shall protect the TSF data from *disclosure, modification* by **verifying encryption and message integrity** when the TSF data is transmitted among TOE’s separated parts.

5.1.6.2. FPT_PST.1 Basic protection of stored TSF data (Extended)

Hierarchical to	No other components.
Dependencies	No dependencies.

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

[TSF data:

- Encryption policy: Table name, column name, user Data Encryption Key (DEK), IP
- Audit data

- Administrator (Super manager, Security user) account information
- Agent account information
- User DB access account information
- D'Guard KMS's token file (Master Key, RSA private key/ public key)
- Agent's token file (RSA private key/ public key)
- Set value of the TOE: Integrity check cycle, IP, Port information, Email address etc.

]

5.1.6.3. FPT_TST.1 TSF Self-test

Hierarchical to No other components.

Dependencies No dependencies.

FPT_TST.1.1 The TSF shall run a suite of self-tests *during initial start-up, periodically during normal operation* to demonstrate the correct operation of [*[Table 25] Items subject to TOE self-test*]

[Table 25] Items subject to TOE self-test

Classification	Item	Content (role)
D'Guard KMS	Cryptographic module	Self-test
	Process(Policy Service)	Process name: java Determine whether the startup was successful during startup or periodically operating during regular operation and generate audit logs
D'Guard Plug-In	Cryptographic module	Self-test
	Process(Agent Daemon)	Process name: eap Determine whether the startup was successful during startup or periodically operating during regular operation and generate audit logs
D'Guard API	Cryptographic module	Self-test

	Process(Agent Daemon)	Process name: java Determine whether the startup was successful during startup or periodically operating during regular operation and generate audit logs

FPT_TST.1.2 The TSF shall provide **authorized administrators** with the capability to verify the integrity of [*“item” of configuration file for [Table 26] Items subject to TOE integrity test*].

FPT_TST.1.3 The TSF shall provide **authorized administrators** with the capability to verify the integrity of [*“item” of Stored TSF executable code for [Table 26] items subject to TOE integrity test*].

Application notes

- ✓ The D’Guard KMS: The period of self-testing periodically during normal operation is 1 hour
- ✓ The D’Guard Plug-In: The period of self-testing periodically during normal operation is 1 hour
- ✓ The D’Guard API: The period of self-testing periodically during normal operation is 1 hour.

[Table 26] Items subject to TOE integrity test

Classification	Item	Content (role)	File
D’Guard KMS	Configuration file	TOE configuration file	dguard.xml db.properties
	Stored TSF executable code	KMS daemon process	propolicyCore.jar pstool.jar
		KCMVP	MagicJCrypto-v3.0.0.jar
D’Guard Plug-In	Configuration file	TOE configuration file	eap.conf license.conf token.conf
	Stored TSF executable	Plug-In behavior	eap eaptoken

	code	process	dgfile dgsam
		Library module	libDGuardAPI_v5.0.1_Linux_4.18_x64.so libDGuardAPI_S_v5.0.1_Linux_4.18_x64.so libDGuardExtproc_v5.0.1_Linux_4.18_x64.so libDGuardExtproc_S_v5.0.1_Linux_4.18_x64.so libDGuardFile_v5.0.1_Linux_4.18_x64.so libDGuardFile_S_v5.0.1_Linux_4.18_x64.so libDGuardEAP_v5.0.1_Linux_4.18_x64.so libINISAFE_PKI_for_C_v5.2.0_Linux_2.6_64.so
		KCMVP	libINISAFE_Crypto_for_C_v5.4.0_Linux_4.18_64.so
D'Guard API	Configuration file	TOE configuration file	dguard.conf
	Stored TSF executable code	Library module	DguardAPI.jar
		KCMVP	MagicJCrypto-v3.0.0.jar

5.1.7. TOE access

5.1.7.1. FTA_MCS.2 Per user attribute limitation on multiple concurrent sessions

Hierarchical to	FTA_MCS.1 Basic limitation on multiple concurrent sessions
Dependencies	FIA_UID.1 Timing of identification

FTA_MCS.2.1 The TSF shall restrict the maximum number of concurrent sessions [belonging to the same **administrator** according to the rules for the list of management functions defined in FMT_SMF1.1]

a) Limit the maximum number of concurrent sessions to 1 for management access by the same administrator who has the right to perform FMT_MOF.1.1 "Management actions" and FMT_MTD.1.1 "Management"

b) Limit the maximum number of concurrent sessions to {1} for management access by the same administrator who doesn't have the right to perform FMT_MOF.1.1 "Management actions" but has the right to perform a query in FMT_MTD.1.1 "Management" only

c) [None]

FTA_MCS.2.2	The TSF shall enforce a limit of [1] session per administrator by default.

5.1.7.2. FTA_SSL.5 Management of TSF-initiated sessions (Extended)

Hierarchical to	No other components.
Dependencies	FIA_UAU.1 authentication or No dependencies.
FTA_SSL.5.1	The TSF shall [*terminate*] the administrator's interactive session after a [*time interval of the administrator inactivity (10 minutes)*]

5.1.7.3. FTA_TSE.1 TOE session establishment

Hierarchical to	No other components.
Dependencies	No dependencies.
FTA_TSF.1.1	The TSF shall be able to refuse the **management access session of the administrator**, based on [Access IP, *the status of activating the management access session of the administrator other than security manager having the same rights*, [None]].

Application notes

- ✓ The number of super manager connectable IPs provided by the TOE is one.

5.2. Security assurance requirements

Assurance requirements of this ST are comprised of assurance components in CC part 3, and the evaluation assurance level is EAL1+. The following table summarizes assurance components.

[Table 27] Security assurance requirements

Security assurance class	Security assurance component	
Security Target evaluation	ASE_INT.1	ST introduction
	ASE_CCL.1	Conformance claims
	ASE_OBJ.1	Security objectives for the 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	Preparative procedures
Life-cycle support	ALC_CMC.1	Labelling of the TOE
	ALC_CMS.1	TOE 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 Introduction

Dependencies	No dependencies.
Developer action elements	
FTA_TSF.1.1	The developer shall provide an ST introduction.
Content and presentation elements	

ASE_INT.1.1C	The ST introduction shall contain an 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 claims

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 that identifies the version of the CC to which the ST and the TOE claim conformance.
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 requirement packages to which the ST claims conformance.
ASE_CCL.1.6C	The conformance claim shall describe any conformance of the ST to a package as either package-conformant or package-augmented.
ASE_CCL.1.7C	The conformance claim rationale shall demonstrate that the TOE type is consistent with the TOE type in the PPs for which conformance is being claimed.
ASE_CCL.1.8C	The conformance claim rationale shall demonstrate that the statement of the security problem definition is consistent with the statement of the security problem definition in the PPs for which conformance is being claimed.
ASE_CCL.1.9C	The conformance claim rationale shall demonstrate that the statement of security objectives is consistent with the statement of security objectives in the PPs for which conformance is being claimed.
ASE_CCL.1.10C	The conformance claim rationale shall demonstrate that the statement of security requirements is consistent with the statement of security requirements in the PPs for which conformance is being claimed.
Evaluator action elements	
ASE_CCL.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.1.3. ASE_OBJ.1 Security objectives for the operational environment

Dependencies	No dependencies.
Developer action elements	
ASE_OBJ.1.1D	The developer shall provide a statement of security objectives
Content and presentation elements	
ASE_OBJ.1.1C	The statement of security objectives shall describe the security objectives for the operational environment

Evaluator action elements	
ASE_OBJ.1.1E	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, classes, and methodology as a model for presentation.
ASE_ECD.1.5C	The extended components shall consist of measurable and objective elements such that conformance or nonconformance 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 requirements 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_RREQ.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 Basic 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.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.1.2E	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 changing 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 of 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 TOE 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 for 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 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 describe 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 procedure
Developer action elements	
ATE_IND.1.1D	The developer shall provide the TOE for testing.
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 procedure
Developer action elements	
AVA_VAN.1.1D	The developer shall provide the TOE for testing.
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 possessing Basic attack potential.

5.3. Security requirements rationale

5.3.1. Dependency rationale of security functional requirements

The following table shows dependency of security functional requirements.

[Table 28] Rationale for the dependency of the security functional requirements

No.	Security functional requirements	Dependency	Reference No.
1	FAU_ARP.1	FAU_SAA.1	3
2	FAU_GEN.1	FPT.STM.1	Rationale (1)
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.3	FAU_STG.1	Rationale (2)
7	FAU_STG.4	FAU_STG.1	Rationale (2)
8	FCS_CKM.1(1)	[FCS_CKM.2 or FCS_COP.1]	11, 13
		FCS_CKM.4	12
9	FCS_CKM.1(2)	[FCS_CKM.2 or FCS_COP.1]	11, 13
		FCS_CKM.4	12
10	FCS-CKM.2	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	9, 10
		FCS_CKM.4	12
11	FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	9, 10
12	FCS_COP.1(1)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	9
		FCS_CKM.4	12
13	FCS_COP.1(2)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	10
		FCS_CKM.4	12
14	FCS_RBG.1	-	-
15	FDP_UDE.1	FCS_COP.1	13, 14
16	FDP_RIP.1	-	-
17	FIA_AFL.1	FIA_UAU.1	21
18	FIA_IMA.1	-	-
19	FIA_SOS.1	-	-
20	FIA_UAU.2	FIA_UID.1	24
21	FIA_UAU.4	-	-
22	FIA_UAU.7	FIA_UAU.1	21
23	FiA_UID.2	-	-
24	FMT_MOF.1	FMT_SMF.1	28
		FMT_SMR.1	29
25	FMT_MTD.1	FMT_SMF.1	28
		FMT_SMR.1	29
26	FMT_PWD.1	FMT_SMF.1	28
		FMT_SMR.1	29
27	FMT_SMF.1	-	-
28	FMT_SMR.1	FIA_UID.1	24
29	FPT_ITT.1	-	-
30	FPT_PST.1	-	-
31	FPT_TST.1	-	-
32	FTA_MCS.2	FIA_UID.1	24
33	FTA_SSL.5	FIA_UAU.1	21
34	FTA_TSE.1	-	-

Rationale (1): FAU_GEN.1 has a dependent relationship with FPT_STM.1, but this security target specification assumes the use of a trusted timestamp and satisfies the dependent relationship in the OE.TIME_STAMP.

Rationale (2): FAU_STG.3 and FAU_STG.4 have the dependency on FAU_STG.1. However, This ST satisfies the dependent relationship in OE.SECURE_DBMS, because the audit data is stored in the DBMS that interacts with the TOE.

FIA_AFL.1, FIA_UAU.7 have FIA_UAU.1 as a dependent relationship, but are satisfied by FIA_UAU.2, which has a hierarchical relationship with FIA_UAU.1.

FIA_UAU.2, FMT_SMR.1, and FTA_MCS.2 have FIA_UID.1 as a dependent relationship, but are satisfied by FIA_UID.2, which has a hierarchical relationship with FIA_UID.1.

5.3.2. Dependency rationale of security assurance requirements

The dependency of EAL1 assurance package provided in the CC is already satisfied, the rationale is omitted.

The augmented SAR ATE_FUN.1 has dependency on ATE_COV.1 but, ATE_FUN.1 is augmented to require developer testing in order to check if the developer correctly performed and documented the tests in the test documentation, ATE_COV.1 is not included in this ST since it is not necessarily required to show the correspondence between tests and the TSFIs.

6. TOE Summary Specification

This chapter describes the security functions that satisfy the SFRs of the TOE

TOE security functionality can be largely divided into security audit, cryptographic support, user data protection, security management, protection of the TSF, and TOE access. Afterwards, this section describes how the TOE satisfies the SFRs specified in [Table 11].

6.1. Security audit (FAU)

When audit data is generated using a reliable timestamp provided by the TOE operation environment, audit data is guaranteed to be generated sequentially.

6.1.1. FAU_ARP.1 Security alert

When the TOE detects a potential security violation, it sends an alert mail to the authorized administrator.

For related potential security violations, see 6.1.3.

When a potential security violation is detected, an alert mail is sent to the mail server, mail server account and mail recipient (authorized administrator) information registered in the configuration file. Alarm mail is sent by D'Guard KMS, and is sent through a common module used internally. One or more mail recipients are registered.

6.1.2. FAU_GEN.1 Audit data generation

The TOE stores audit data generated during operation.

Audit data generated by D'Guard KMS is immediately stored in the audit data store.

The audit data generated by the agent is collected from the log daemon of D'Guard KMS, first saved as a file, and then stored in the audit data store. The files collected from the log daemon are stored in the audit data store at regular intervals (5 minutes).

The following audit data is generated in the TOE.

[Table 29] Audit data type

Subject	Audit data
D’Guard KMS	Identification and authentication, shutdown of session
	Registration, modification and deletion of the policy management
	Start-up and shutdown of the service
	Master security token creation and super manager password initialization
	Self-testing
	Policy distribution
	In case of potential security violation, audit data is generated after sending an alert email
D’Guard Plug-In	Start-up and shutdown of the service
	Policy distribution and mutual authentication
	User data encryption
	Self-testing
D’Guard API	Policy distribution and mutual authentication
	User data encryption
	Self-testing

The audit data generated by the TOE includes the following.

[Table 30] Audit data composition

Composition	Content
Time	· Audit data creation time
Subject	· D’Guard KMS (Policy, Security Admin, PSTOOL) · Agent hos name
Type	· Administrator log-in and log-out · Start-up and shutdown of the service · Registration, modification and deletion of the security management items (Including modified TSF data value) · Management and distribution of the policy · Agent user data encryption/decryption · TSF data or execution code changed in case of integrity violation · Self-testing etc.
Success or Failure	· Results of audit data success/failure

content	· Audit data details

6.1.3. FAU_SAA.1 Potential violation analysis

When the TOE detects a potential violation event such as a self-test of the cryptographic module, the TOE sends an alert email to the authorized administrator.

[Table 31] Potential violation events and Action

Security violation events	Action
Accumulation of authentication failure events (5 times)	· Sending a warning emails to authorized administrators · Limitation on login attempts for 5 minutes (disable authentication)
Self-test failure	· Sending a warning emails to authorized administrators
Disk capacity excess (80%)	· Sending a warning emails to authorized administrators
Disk capacity saturation (90%)	· Sending a warning emails to authorized administrators · Deletion on some of the old audit records (5%)

6.1.4. FAU_SAR.1 Audit review

The TOE provides the authorized administrator with a function to review all the audit data of the TOE according to the subject of the event, the type of audit data, and the time of the audit data creation. Audit review is provided by the GUI function of D'Guard KMS. Super managers and security users have the authority to review audits.

For the types of TOE audit data, see [Table 29].

6.1.5. FAU_SAR.3 Selectable audit review

The TOE orders in descending order based on the occurrence time of the audit data based on the selection criteria according to the events to be audited.

The search conditions for all audit data by the authorized manager can be viewed by AND operation when entering the "Selection Criteria" value.

6.1.6. FAU_STG.3 Action in case of possible audit data loss

The TOE provides a function to action and prevent audit data loss. When the space of the audit trail storage reaches the specified limit, a security alert mail is sent to the authorized administrator.

The maximum size of the audit trail repository is not limited by the TOE and is determined by the size of the system repository.

If the audit storage exceeded 80%, the alert log is sent to the authorized administrator after generating the audit log.

6.1.7. FAU_STG.4 Prevention of audit data loss

When the audit storage saturation stage exceeds 90%, the audit log corresponding to 5% of the entire audit record is deleted with the sending of an alert email to the authorized administrator after the generation of the audit log.

6.1.8. SFR Mapping

SFR to be satisfied: FAU_ARP.1, FAU_GEN.1, FAU_SAA.1, FAU_SAR.1, FAU_SAR.3, FAU_STG.1, FAU_STG.3, FAU_STG.4

6.2. Cryptographic support (FCS)

The TOE provides encryption functions for TSF data and user data. In addition, it provides functions such as generating, distributing, and destroying encryption keys necessary to provide encryption functions.

For the validated cryptographic modules that have been used in the TOE, see [Table 8] The validated cryptographic module of the TOE.

6.2.1. FCS_CKM.1(1) Cryptographic key generation (User data encryption)

The TOE generates an encryption key for encrypting user data using the verification algorithm of the validated cryptographic module. The identification information of the validated cryptographic module is as follows. For information such as the verification number, refer to [Table 8] The validated cryptographic module of the TOE.

The validated cryptographic module used in D'Guard KMS is MagicJCrypto, and the identification modules are MagicJCrypto-v3.0.0.jar. The validated cryptographic module used in D'Guard Plug-In is INICrypto, and the identification module is libINISAFE_Crypto_for_C_v5.4.0_Linux_4.18_64.so. The validated cryptographic module used in D'Guard API is MagicJCrypto, and the identification modules are MagicJCrypto-3.0.0.jar.

The user data encryption algorithm supported by the TOE supports ARIA128, ARIA256, LEA128, LEA256 and SEED128 of symmetric key encryption. The length of the encryption key is 128 bits for ARIA128/LEA128/SEED128, 256 bits for ARIA256/LEA256, depending on the encryption ratio. It also supports secure hash SHA256 and SHA512 algorithms for encryption that does not decrypt, such as passwords. In the case of secure hash, a 256-bit encryption key is generated and used as a salt when using the algorithm.

For more information on the algorithm used for encrypting user data, the secret ratio of the encryption key and the supported algorithm, refer to [Table 15] User data encryption algorithm and key length.

6.2.2. FCS_CKM.1(2) Cryptographic key generation (TSF data encryption)

The TOE generates an encryption key for TSF data encryption using the verification algorithm of the validated cryptographic module. The identification information of the validated cryptographic module is as follows. For information such as the verification number, refer to [Table 8] The validated cryptographic module of the TOE.

The validated cryptographic module used in the TOE is the same as FCS_CKM (1).

The encryption key generation for TSF data encryption in the TOE includes the master key for encryption of security policy and audit log, session key for encryption of communication section, RSA public key pair for mutual authentication and session key exchange, and KEK (Key for encryption of encryption key).

The master key and session key are generated by a random bit generator. The algorithm uses HASH-DRBG-SHA256, and the standard list is ISO/IEC 18031. The actual encryption uses the ARIA256 algorithm, and the encryption key length is 256 bits.

The RSAES algorithm is used for RSA public key pair generation for mutual authentication and session key exchange, and the standard list is ISO / IEC 18033-2. The actual encryption uses the RSAES algorithm, and the encryption key length is 2048 bits.

KEK generation uses the PBKDF2 algorithm, and the standard list is PKCS # 5 RFC 2898. The actual encryption uses the ARIA256 algorithm, and the encryption key length is 256 bits.

For detailed information on the algorithm used for TSF data encryption, the encryption key ratio and the supported algorithm, refer to [Table 16] TSF data cryptographic key algorithm and key sizes.

6.2.3. FCS_CKM.2 Cryptographic key distribution

The TOE distributes the cryptographic key through the cryptographic key distribution method defined in FCS_CKM.2.1. The algorithm used for the distribution of the cryptographic key uses the cryptographic algorithm to be verified by the validated cryptographic module that has been verified for safety and implementation suitability. For the standard list used for distribution of cryptographic keys, see [Table 17] Cryptographic key distribution method.

As a preliminary task for distribution of cryptographic keys, D'Guard KMS generates a security token to be used by the agent. The security token is generated in the form of a file for delivery to the agent. The agent security token contains the public key pair for the session key exchange, the ID of the security token, and the password of the security token, and is encrypted using PBKDF2.

The cryptographic key distribution procedure is as follows.

- The agent generates a part of the first session key using a random bit generator and the first session key is transmitted by encrypting it with the public key of D'Guard KMS.
- D'Guard KMS decrypts the first session key with a secret key, generates a part of the second session key using a random bit generator, encrypts and sends the second session key with the agent's public key.
- The agent decrypts the second session key with a secret key and combines a part of the first session key generated by the agent with a part of the second session key generated by D'Guard KMS. After that, generate the third session key and send the result to D'Guard KMS
- The third session key is generated by combining the second session key of D'Guard KMS and the first session key of the agent, and the third session key is exchanged and completed
- D'Guard KMS encrypts the user data encryption key and security policy with the third session key and transmits it to the agent.
- The agent decrypts the user data encryption key and security policy with the third session key and completes distribution of the encryption key.
- After the session key is exchanged, data for data integrity verification is transmitted together when the encryption key is distributed. Data for integrity verification is generated using HMAC-SHA256 for the entire professional data.

6.2.4. FCS_CKM.4 Cryptographic destruction

D'Guard KMS destroys the cryptographic key for encryption of user data generated and managed by the super manager when executing the SQL statement when the cryptographic key is deleted from the management screen and deletes it from the storage (DB).

In memory area zeroing, the actual memory area is overwritten three times with 0x00 to destroy existing data.

The master key loaded and used in memory is destroyed by zeroing the memory area at the end of the process.

The session key loaded and used in memory is destroyed by zeroing the memory area at the end of communication.

The security policy information loaded in the memory for distribution to the agent is destroyed by zeroing the memory area after completing the policy distribution communication.

The cryptographic key for encrypting user data used by the agent (D'Guard Plug-In, D'Guard API) exists only in the memory area. After the cryptographic operation is completed, the memory area is zeroed and destroyed.

The cryptographic key for encrypting the security policy information distributed to the agent exists in the memory area, and when the process ends, the memory area is zeroed and destroyed. The session key loaded and used in memory is destroyed by zeroing the memory area at the end of communication.

For information on the cryptographic key and the destruction method to be destroyed, refer to [Table 18] Cryptographic key destruction method of D'Guard KMS and [Table 19] Cryptographic key destruction method of Agent.

6.2.5. FCS_COP.1(1) Cryptographic operation (User data encryption)

The TOE performs user data cryptographic operation using the cryptographic algorithm to be verified by the validated cryptographic module that has been verified for safety and implementation conformity through the cryptographic module verification system, and the validated cryptographic module that is verified uses the operation mode for verification when performing cryptographic operation.

The validated cryptographic module used in the TOE is the same as FCS_CKM (1).

The user data encryption algorithm supported by the TOE supports ARIA128, ARIA256, LEA128, LEA256 and SEED128 of symmetric key encryption. It also supports secure hash SHA256 and SHA512 algorithms for encryption that does not decrypt, such as passwords.

For the symmetric-key encryption ARIA128 and ARIA256 algorithms, the standard list is KSX 1213-1 in 128-bit and 256-bit encryption key length. For the LEA128 and LEA256 algorithms, the standard list of 128-bit and 256-bit encryption keys is TTAK.KO-12.0223. For the SEED128 algorithm, the standard list of 128-bit encryption keys is TTAS.KO-12.0004/R1.

The standard list of Secure Hash SHA256 and SHA512 algorithms is ISO / IEC 10118-3. The length of the generated cryptographic key is 256 bits, which is used as salt in the actual cryptographic operation.

For more information on the algorithm used for encrypting user data, encryption key ratio and supported algorithm, refer to [Table 15] User data encryption algorithm and key length.

6.2.6. FCS_COP.1(2) Cryptographic operation (TSF data encryption)

The TOE performs TSF data cryptographic operation using the cryptographic algorithm to be verified by the validated cryptographic module that has been verified for safety and implementation conformity through the cryptographic module verification system, and the validated cryptographic module that is verified uses the operation mode for verification when performing cryptographic operation.

The validated cryptographic module used in the TOE is the same as FCS_CKM (1).

D'Guard KMS uses the ARIA256 algorithm to encrypt the token file, security policy and audit log, key information of the configuration file, communication section and key information loaded in memory, and the standard list for the length of the 256-bit cryptographic key is KSX 1213-1.

The one-way encryption of the administrator password uses the SHA256 algorithm, and the standard list is ISO/IEC 10118-3. RASAE algorithm is used for mutual authentication and session key exchange, and the standard list of 2048-bit cryptographic key length is ISO/IEC 18033-2.

The encryption of security policy, communication section and user data encryption key loaded in memory in the agent (D'Guard Plug-In, D'Guard API) use ARIA256 algorithm, and the standard list for 256-bit cryptographic key length is KSX 1213-1. The RASAE algorithm is used for mutual authentication and session key exchange, and the standard list of 2048-bit cryptographic key length is ISO/IEC 18033-2.

For TSF data encryption target, cryptographic algorithm, and key length information, refer to [Table 16] TSF data cryptographic algorithm and key sizes.

6.2.7. FCS_RBG.1 Random bit generation (Extended)

The TOE uses a secure random bit generator of the validated cryptographic modules whose safety and implementation suitability are verified through the cryptographic module verification system. The TOE does not allow the use of the non-verified algorithm when using the validated cryptographic module.

The TOE uses the HASH-DRBG-SHA256 random bit generator when generating the cryptographic key, and the key length generated through the random bit generator is the default value of 32 bytes. The key is generated as long as the TOE is input as an argument (multiple positive number of 8). If the length according to the cryptographic ratio is entered as a factor, a key of the corresponding length is generated.

6.2.8. SFR Mapping

SFR to be satisfied: FCS_CKM.1(1), FCS_CKM.1(2), FCS_CKM.2, FCS_CKM.4, FCS_COP.1(1), FCS_COP.1(2), FCS_RBG.1(Extended)

6.3. User data protection (FDP)

In order to protect user data, the TOE encrypts it using the validated cryptographic module that has been verified for its stability and implementation suitability, and deletes some residual information used internally.

6.3.1. FDP_UDE.1 User data encryption

The TOE provides encryption/decryption functions for each column to protect user data.

The API type supports encryption/decryption of user data by calling the encryption interface (encryption, decryption API, etc.) from the user application, and the Plug-In type supports encryption/decryption by calling the encryption interface from the DBMS.

The user data encryption processing procedure is as follows.

Plug-In Type

- The authorized administrator sets the cryptographic policy, such as generating a security token and generating a cryptographic key to be used by D'Guard KMS and agents.
- The authorized administrator manually distributes the security token to be used by the agent to the agent system.
- The agent receives the cryptographic key and security policy using the distributed security token. The agent encrypts and stores it in the shared memory inside the system.
- When the user calls the initialization interface for encryption in the Oracle Database, the agent cryptographic module checks the information of the agent security token, decrypts the cryptographic policy stored in the shared memory in the system, and encrypts and stores it in the internal memory.
- The agent cryptographic module encrypts user data with the cryptographic policy information in the internal memory when the user calls the encryption interface from the Oracle Database.

API Type

- The authorized administrator sets the cryptographic policy, such as generating a security token and generating a cryptographic key to be used by D'Guard KMS and agents.
- The authorized administrator manually distributes the security token to be used by the agent to the agent system.
- When the Java interface (WAS, etc.) calls the initialization interface to encrypt the Java API interface, the agent receives the cryptographic key and security policy using the distributed security token. And, the agent encrypts and stores the cryptographic key and security policy in the internal memory.
- When the Java interface (WAS, etc.) calls the encryption interface of the Java API interface, the agent encrypts the user data with cryptographic policy information in the internal memory.

The TOE provides the basic option that the same encrypted text is not generated for the same plain text when the authorized administrator generates the cryptographic key for user data in D'Guard KMS.

The internal procedure of generating different ciphertexts for the same plaintext is as follows.

Encryption using random bit IV

- The random bit generator is used to generate a 6-byte random number.
- Combining the IV 16 bytes of the cryptographic policy received from the D'Guard KMS and 6 bytes of the random number, a random bit IV is generated with 16 bytes out of the 32 bytes values generated through the SHA256 algorithm.
- User data encryption key and random bit IV of the cryptographic policy received from D'Guard KMS encrypt user data.
- After combining the 6-byte random number and the user data encryption result, base64 encoding is performed to complete the encryption.

Decryption using random bit IV

- The user data ciphertext is Base64 decoded.
- Among the Base64 decoded bytes, the front 6 bytes are extracted with a random number.
- Combining the IV 16 bytes of the cryptographic policy received from the D'Guard KMS and the front 6 bytes of the random number, a random bit IV is generated with 16 bytes out of the 32 bytes values generated through the SHA256 algorithm.
- User data encryption key and random bit IV of the cryptographic policy received from D'Guard KMS decrypt user data.

6.3.2. FDP_RIP.1 Subset residual information protection

The TOE satisfies the requirement that all original user data is deleted after encryption/decryption of the user data.

The Deletion of original user data refers to unrecoverable deletion. In the case of the Plug-In type and API type of the TOE, encryption and decryption of user data is performed through user application development or modification. In the case of the Plug-In type, the query included in the application code is modified and applied. In the case of the API type, the application code is directly modified and applied. The TOE does not store the user data to be encrypted separately, and the TOE documentation includes precautions so that the application developer can delete all the original user data in compliance with the requirements provided by the TOE.

6.3.3. SFR Mapping

SFR to be satisfied: FDP_UDE.1, FDP_RIP.1

6.4. Identification and authentication (FIA)

The TOE identifies the authorized administrator accessing the management server. No functions of the TOE can be used until identification is achieved.

6.4.1. FIA_AFL.1 Authentication failure handling

The TOE provides the user account lock function to protect the TOE from malicious user authentication attempts. When the number of failed authentication attempts exceeds the set information, the TOE performs identification and deactivation of authentication functions for a certain period of time.

If the authentication failure deactivation count (5 times) is exceeded, the account is deactivated (5 minutes) and authentication cannot be attempted, and an alert email is sent to the authorized administrator.

6.4.2. FIA_IMA.1 TOE internal mutual authentication (Extended)

The TOE performs mutual authentication between the physically separated TOE components through its internally implemented authentication protocol, and the TOE components are D'Guard KMS and agents (D'Guard Plug-In, D'Guard API).

The RASAE algorithm is used for mutual authentication between TOE components, and the standard list of 2048-bit encryption key length is ISO/IEC 18033-2.

The time when mutual authentication is performed in the TOE is performed when the agent starts a service and requests security policy distribution to D'Guard KMS. The mutual authentication procedure and mechanism are as follows.

- In advance, D'Guard KMS generates a security token (Agent public key pair, D'Guard KMS public key) to be used by the agent and manually distributes it to the agent.
- After generating a part of the first session key from the agent to the random bit generator, it is encrypted and transmitted with the public key of D'Guard KMS.

- The secret key of D'Guard KMS decrypts the first session key. The random number generator of D'Guard KMS generates part of the second session key, and the second session key is encrypted with the agent's public key and transmitted.
- The agent's secret key decrypts the second session key. Part of the first session key generated by the agent and part of the second session key generated by D'Guard KMS are combined to generate a third session key. After that, the result is sent to D'Guard KMS.
- D'Guard KMS checks the result of generating the third session key. Part of the first session key generated by the agent and part of the second session key generated by D'Guard KMS are combined to generate a third session key. Exchange the third session key and complete the process.
- D'Guard KMS encrypts the policy information including the encryption key with the third session key that has been exchanged and sends it to the agent.
- The agent decrypts the policy information including the encryption key with the third session key that has been exchanged.

6.4.3. FIA_SOS.1 Verification of secrets

The TOE is operated so that the confidential information satisfies the acceptance criteria defined by the authorized administrator. The password of the authorized administrator is combined with three or more rules among alphabet / number / special characters, and is generated with a length of at least 10 digits and a maximum of 20 digits.

The TOE verifies that the following defined allowance criteria for the administrator password are satisfied at the time of password registration and change.

Minimum/Maximum length

- 10 to 20 digits

Combination rules

- Combination of 3 or more among English alphabets/ numbers / special characters
- Excluded if a password has 3 or more successive letters (abc, def, 123, 432, etc.)
- Excluded if a password has 3 or more successive letters on the keyboard (asd, qwe, jkl, etc.)
- Excluded if a password has 3 or more same letters (aaa, 222, 999, etc.)

- Excluded if a password contains administrator account information
- Excluded if a password was used the last three times (3 times by default)

Change interval (the period during which the password is used)

· 60 days by default, the interval is defined by the authorized administrator

When the D'Guard KMS is accessed for the first time, the super manager is provided with the default (account: manager, password: admin) information. Upon authentication and initialization, the super manager can change the account information. Should be changed.

6.4.4. FIA_UAU.2 User authentication prior to all actions

The security administrator (super manager, security user) can perform web security management after D'Guard KMS authentication through password based on administrator ID information through the screen interface.

Super manager can perform console security management after D'Guard KMS authentication through password based on administrator ID information.

6.4.5. FIA_UAU.4 Single-use authentication mechanism

The security administrator (super manager, security user) generates a security character (Captcha) for each session on the login screen of the web GUI environment and verifies it by entering it by the security administrator. Re-use of security administrator authentication is prevented through verification of the security letter. The valid period of the captcha is 60 seconds, and it is a 5-digit character composed of lowercase letters and numbers.

The super manager verifies the security code displayed on the console screen of the CLI environment for each session and inputs it to the super manager. The security code has a validity period of 60 seconds and consists of 5 digits.

6.4.6. FIA_UAU.7 Protected authentication feedback

The web authentication: The replacement character ‘*’ is displayed on the password input screen, and ‘The Login information does not match’ is displayed on the screen when authentication fails.

The console authentication: The Password input character is not displayed, and if authentication fails, 'Login Failed' is displayed on the screen.

6.4.7. FIA_UID.2 User identification prior to all actions

Security administrators (super manager, security users) can perform web security management after D'Guard KMS authentication through password based on security manager ID information through a screen interface.

Super manager can perform console security management after D'Guard KMS authentication through password based on administrator ID information.

6.4.8. SFR Mapping

SFR to be satisfied: FIA_AFL.1, FIA_IMA.1(Extended), FIA_SOS.1, FIA_UAU.2, FIA_UAU.4, FIA_UAU.7, FIA_UID.2

6.5. Security management (FMT)

The security management of the TOE provides security management functions such as response actions, rules, and audits.

6.5.1. FMT_MOF.1 Management of security functions behavior

The TOE performs security function management such as adding, deleting, and modifying condition rules that can determine the security function behavior. Security function management restricts the role to be performed by an authorized administrator.

An authorized administrator means a security administrator. The security administrator is divided into a super manager and a security user. In the case of D'Guard KMS web GUI environment, both the super manager and the security user perform security function management, and only the super manager has the ability to stop, start, and modify the behavior. In the case of the console CLI environment, only the super manager can manage the security functions.

[Table 32] Menu of the web management server

Menu	Sub menu	Description

Account	Token management	Security token for agent assignment
	User management	Management of the super manager and security user
Key	Key management	Generate and manage user data encryption keys
DB Encryption	DBMS management	Management of the database information
	Policy management	Management of encryption policies and permissions
Audit	Audit log	Search for audit data
Configuration	User constraints	Constraints management for administrators and security users
	Server configurations	Configurations of the D’Guard KMS

[Table 33] Menu of the console management server

Menu	Description
Operation of the service	Start-up/shut down of the D’Guard KMS’s service
Encryption	Encrypt important data information in the D’Guard KMS's configuration file
Integrity	Generate integrity information stored in configuration files

6.5.2. FMT_MTD.1 Management of TSF data

The TOE provides management donation for TSF data. In addition, it is possible to perform the functions of modification, deletion, and generation other than query according to the role, and restricts the execution role to an authorized administrator.

The authorized administrator's ability to manage TSF data is as follows.

The super manager initializes the super manager information for starting and operating the TOE in the console CLI environment, and initializes the security token including the public key pair and master key of D'Guard KMS.

[Table 34] List of TSF data and management ability

Administrator Type	TSF data	Ability				
		Query	Modify	Delete	Generate	Initialize
Security user	Account information	O	-	-	-	-
	Encryption policy	O	-	-	-	-
	Audit data	O	-	-	-	-
	Configuration information	-	-	-	-	-
Super manager (Web GUI)	Account information	O	O	O	O	-
	Encryption policy	O	O	O	O	-
	Audit data	O	-	-	-	-
	Configuration information	O	O	O	O	-
Super manager (Console CLI)	Account information	-	-	-	-	O
	Encryption policy	-	-	-	-	-
	Audit data	-	-	-	-	-
	Configuration information	-	O	-	-	O

6.5.3. FMT_PWD.1 Management of ID and password (Extended)

The creation and modification of accounts and passwords is limited to authorized administrators.

The password is limited to a character length between 10 and 20 digits in 3 combinations including alphabetic characters, numbers and special characters. The TSF also forces the password to be initialized when the TOE is accessed for the first time in the console CLI environment during the installation process. The password is encrypted and stored using the SHA256 algorithm in the D'Guard KMS internal DB. Refer to FIA_SOS.1 for the password length and combination rules selected when registering or changing the password.

TSF forces the password to be initialized during the first TOE access in the console CLI environment during the installation process. The password is encrypted and stored in the D'Guard KMS internal

DB using the SHA256 algorithm.

6.5.4. FMT_SMF.1 Specification of Management Functions

The TOE performs the management function of the security attribute list specified in FMT_MOF.1 / FMT_MTD.1 / FMT_PWD.1.

The authorized administrators are restricted from using TOE security functions according to their management capabilities. The super manager can use all security functions provided by the TOE, but the security user only has the ability to decide the action in addition to the ability to stop, initiate, and modify the action. For the security functions provided by the TOE and whether the security functions of the administrator are available, refer to [Table 23] List of the security functions behavior of administrator.

The TOE limits TSF data management according to the role of the authorized administrator. The super manager can manage all TSF data managed by the TOE, and the security user can manage queries only in addition to the TSF data generation, modification, and deletion management functions. TSF data includes security policy, audit log, administrator account information, agent account information, user DB access account information, D'Guard KMS token file, agent token file, and TOE settings.

The TOE limits the administrator's generation function and administrator's password change function to authorized administrators. The super manager can perform the super manager and all security user generation and password change functions. Security users cannot be generated as administrators, only their passwords can be changed.

When generating and changing an administrator's password, it must be set according to the prescribed password combination rules. Combination rules must use 3 combinations of English letters / numbers / special characters with a length of 10 to 20 digits. Consecutive characters must not be more than 3 digits and cannot be used even if the same character is 3 or more times. The administrator account information should not be included, and the password used before and the same password cannot be used up to 3 times.

When installing the TOE, the super manager's ID and password must be set, but the password must be set according to the combination rules.

6.5.5. FMT_SMR.1 Security roles

The TOE generates and initializes a super manager when D'Guard KMS is initialized in the console

CLI environment. The super manager is the only account that holds all rights of D'Guard KMS.

The security user can be generated / modified / deleted by the super manager in the web GUI environment. Security users can only search in the account, policy, and audit menus except for the security settings menu of D'Guard KMS.

6.5.6. SFR Mapping

SFR to be satisfied: FMT_MOF.1, FMT_MTD.1, FMT_PWD.1(Extended), FMT_SFM.1, FMT_SMR.1

6.6. Protection of the TSF (FPT)

The TOE provides a protection function for internally transmitted TSF data.

6.6.1. FPT_ITT.1 Basic internal TSF data transfer protection

The TOE protects TSF data from exposure when TSF data is transferred between separate parts of the TOE. The communication section encryption between D'Guard KMS and the agent uses the HandShake encryption method using the validated cryptographic module to exchange session keys for the section encryption, and performs communication section encryption and mutual authentication between the TOE modules.

D'Guard KMS and the agent have a public key pair, and the public key length is 2048 bits public key. All parameters transmitted after the session key exchange and mutual authentication are completed are encrypted and transmitted using the session key. The transmitted parameters include user data encryption key and authority information as security policy information.

[Table 35] Basic internal TSF data transfer encryption

Classification	Item	Description

Mutual authentication and session key exchange	Mutual authentication (Asymmetric key encryption)	· Generating a security token (including public key pair) for the agent in D'Guard KMS · Manual distribution of the agent's public key pair to the agent · Encrypt the random bit in the agent with D'Guard KMS public key and send it · Decrypt and confirm with own secret key in D'Guard KMS · Encrypt the random bit in the D’Guard KMS with the agent public key and send it · Decrypt and confirm with own secret key in the agent · D'Guard KMS and the agent combine the two random bit exchanged with each other, and then exchange and complete the session key. · The agent sends the result to complete mutual authentication
	Cryptographic algorithm	**Random bit generator** · HASH-DRBG-SHA256 **Integrity** · HMAC-SHA256 **Asymmetric key encryption** · RSAES
Transferred data	Symmetric encryption	· D'Guard KMS sends security policy with session key encryption
	Cryptographic algorithm	**Symmetric encryption** · ARIA256 **Integrity** · HMAC-SHA256

6.6.2. FPT_PST.1 Basic protection of stored TSF data (Extended)

The TOE protects TSF data stored in the storage controlled by the TSF from unauthorized exposure.

Basically, TSF data that is encrypted and stored is as follows.

- ✓ Cryptographic key (User data encryption key, Master key, etc.)
- ✓ Critical security parameters (Password option values such as IV value)
- ✓ TOE settings (Security policy and environment settings)

- ✓ Administrator account information
- ✓ Policy DB account information
- ✓ User DB account information

The TOE does not perform encryption on configuration files or executable files that do not include the TOE configuration values, such as IP address, port information, and security alarm email address. Configuration files under the TOE installation folder, config folder, executable modules under the lib folder, and executable script files under the bin folder are included in the integrity check.

The mechanisms for protecting stored TSF data are data encryption and integrity verification. The cryptographic key and critical security parameters, which are essential encryption targets among TSF data, are protected by storing the integrity information together with encryption.

The security token file of D'Guard KMS is encrypted with ARIA256 algorithm using KEK derived from PBKDF2 using the PIN entered by the administrator. The security token includes the public key pair and master key of D'Guard KMS. The master key is the public key of the D'Guard KMS RSA public key pair, and is second-encrypted by the RSAES algorithm.

The algorithm used to store the administrator password in one-way encryption is SHA256.

When storing the TOE settings, critical information such as IP address is encrypted using the ARIA256 algorithm using the master key.

The cryptographic key and critical security parameters in the agent (D'Guard Plug-In, D'Guard API) are encrypted with ARIA256 algorithm using KEK derived from PBKDF2. PBKDF2 is a password-based key derivation algorithm. The password used as a parameter is a combination of the password declared in the source code and the host name.

The symmetric cryptographic algorithm used in TSF data of the TOE is ARIA256, the cryptographic mode is CBC, and the cryptographic key length is 256 bits.

The critical TSF data stored in the TOE is encrypted using the validated cryptographic module. The following is the TSF data stored in the TOE.

[Table 36] Encryption of stored TSF data

Components	TSF data	Storage location	Protection method

D’Guard KMS	Security token	File	ARIA256-CBC (PBKDF2)
	User data cryptographic key	Policy DB	ARIA256-CBC
	Security Policy (User data encryption)	Policy DB	ARIA256-CBC
	Administrator account information	Policy DB	SHA256
	Policy DB account information	File	ARIA256-CBC
	User DB account information	Policy DB	ARIA256-CBC
	Security token information of Agent	Policy DB	ARIA256-CBC
	Audit data	Policy DB	ARIA256-CBC
	Integrity	Policy DB	HMAC-SHA256
D’Guard Plug-In	Security token	File	ARIA256-CBC (PBKDF2)
	Configuration file	File	ARIA256-CBC
	Configuration file (Module integrity)	File	HMAC-SHA256
D’Guard API	Security token	File	ARIA256-CBC (PBKDF2)
	Configuration file	File	ARIA256-CBC
	Configuration file (Module integrity)	File	HMAC-SHA256

6.6.3. FPT_TST.1 TSF testing

The TOE performs self-tests periodically at startup and during normal operation to verify the correct operation of all TSFs. The self-test performs major functional tests necessary for the operation of the TOE and integrity verification of TSF execution modules and configuration files.

D'Guard KMS performs self-tests periodically at service startup and during regular operation (1 hour). The self-test performs an integrity check by comparing the main functions required for the operation of D'Guard KMS and the integrity values of the TSF execution module and configuration file. Depending on the point of failure of the self-test, an alarm e-mail is sent to the registered manager and then stopped at startup. If it is in operation, an alert email is sent to the registered manager.

D'Guard Plug-In performs self-tests periodically at service startup and during regular operation (1 hour). The self-test performs an integrity check by comparing the main functions required for the operation of the D'Guard Plug-In and the integrity values of the TSF execution module and configuration files. Depending on the point of failure of the self-test, the audit data is transmitted and stopped at startup, and only the audit data is transmitted when in operation.

The D'Guard API runs its own tests periodically (1 hour) at service startup and during normal operation. The self-test performs an integrity check by comparing the main functions required for the operation of the D'Guard API and the integrity values of the TSF execution module and configuration file. Depending on the point of failure of the self-test, when the initialization interface is called from the Java program, the audit data is transmitted and then the exception is handled. When operating, only the audit data is transmitted.

For details of the TOE self-test subject, refer to [Table 25] Items subject to TOE self-test and [Table 26] Items subject to TOE integrity test.

6.6.4. SFR Mapping

SFR to be satisfied: FPT_ITT.1, FPT_PST.1(Extended), FPT_TST.1

6.7. TOE access (FTA)

The TOE provides session setting functions such as limiting the number of sessions per user attribute and session timeout for access to the management server.

6.7.1. FTA_MCS.2 Per user attribute limitation on multiple concurred sessions

The TOE provides a function of limiting the number of basic concurrent access sessions for D'Guard KMS access. The TOE provides a function of limiting the number of basic concurrent access sessions for D'Guard KMS access.

D'Guard KMS stores the administrator session information in memory when the security administrator (super manager, security user) authenticates. Thereafter, when authenticating to the same administrator, memory session information is checked to limit concurrent sessions. In addition, one default concurrent session limit for the same manager is forced.

At the same time when the security administrator (super manager, security user) is connected simultaneously and in the case of the web GUI environment, the previously connected session is forcibly terminated. In addition, in the case of a console CLI environment, simultaneous access by the super manager is blocked from new connections.

6.7.2. FTA_SSL.5 Management of TSF-initiated sessions (Extended)

The TOE automatically terminates the session when the inactivity period of the administrator and security user exceeds the set time. In D'Guard KMS 'web GUI and console CLI session, if there is no interaction after the administrator inactivity period (10 minutes), the session is terminated. Session termination due to inactivity applies to both super administrators and security users.

6.7.3. FTA_TSE.1 TOE session establishment

The TOE provides the function to set the access allowance list based on the access IP address and administrator account.

D'Guard KMS security administrator (super manager, security user) restricts session to the same administrator based on the access IP. In the case of the super manager, it is limited to one IP that can be accessed when the installation is initialized. In the case of security users, the super manager can set up to 2 accessible IPs per administrator account. For reference, the administrator can modify the IP that can be accessed by the super manager, and the previous connection session is automatically blocked when the connection IP is modified.

Only one IP registration that can be accessed by the security administrator can be registered, and settings that mean the entire range of the network are not allowed.

6.7.4. SFR Mapping

SFR to be satisfied: FTA_MCS.2, FTA_SSL.5(Extended), FTA_TSE.1