INITECH
1st Financial & Security Service

Security Target (ST)

SafeDB V4.1

INITECH Co., Ltd.

Revision History

Version	Date	Author	Revision
1.0	Mar. 2, 2023	Ha Minho	Initial release
1.1	Jan. 26, 2024	Jeong Munhui	Cryptographic algorithm changed and details on cryptographic keys supplemented
1.2	Jan. 26, 2024	Ha Minho	Explanation of mandatory programs of the TOE added, and information on 3rd party added
1.3	Feb. 2, 2024	Ha Minho	Information on non-TOE operational environment modified Physical scope of the TOE modified Details on cryptographic algorithms modified
1.4	Mar. 19, 2024	Jeong Munhui	Mandatory SW version modified Review findings incorporated
1.5	May 24, 2024	Ha Minho	Review findings incorporated TOE version changed
1.6	June 07, 2024	Ha Minho	Review findings incorporated Change TOE component version

Table of Contents

List of Figures

List of Tables

1. ST Introduction

This chapter introduces the Security Target (ST) of SafeDB V4.1 of INITECH Co., Ltd.

1.1. ST reference

[Table 1] ST reference

Title	Security Target (ST)
Version	V1.6
Author	Ha Minho of INITECH Co., Ltd.
Publication Date	June 07, 2024
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)
Evaluation Assurance Level	EAL1+ (ATE_FUN.1)
Configuration Management No.	CCP.C_SB41_Security Target (ST)_V1.6
Product Classification	DB Encryption
Keywords	Database, Encryption

1.2. TOE overview

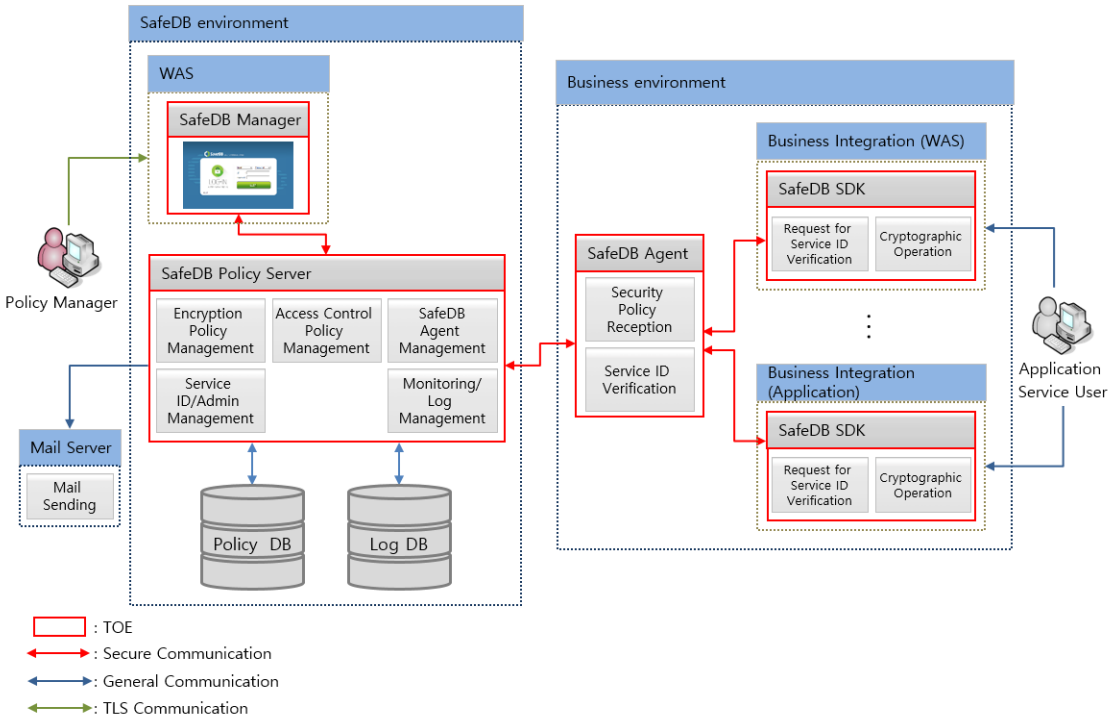
SafeDB V4.1 (hereinafter referred to as the TOE) is a product that uses the standard encryption algorithm and encrypts important information in the database (hereinafter "DB") by the unit of column to make it encrypted and thereby illegible even in the event of illegal disclosure by an insider or an outsider. The TOE has been designed to be safe against hacking attacks, such as sniffing, by encrypting the network among the product modules. In addition, it is equipped with a mechanism to manage keys used for encryption in respect of the security.

The TOE consists of SafeDB Policy Server that manages the cryptographic operation policy and keys; SafeDB Manager that performs the web security management; SafeDB Agent installed and operated on the business server; SafeDB SDK (SafeDB SDK for C, and SafeDB SDK for Java) that supports C/Java language; and SafeDB Plug-In applied to and operated in the DBMS.

If the Security Manager enters the policy information through SafeDB Manager, the security policy is managed by SafeDB Policy Server. The security policy is distributed to SafeDB Agent. The received security policy is referenced during cryptographic operations in SafeDB SDK and SafeDB Plug-In.

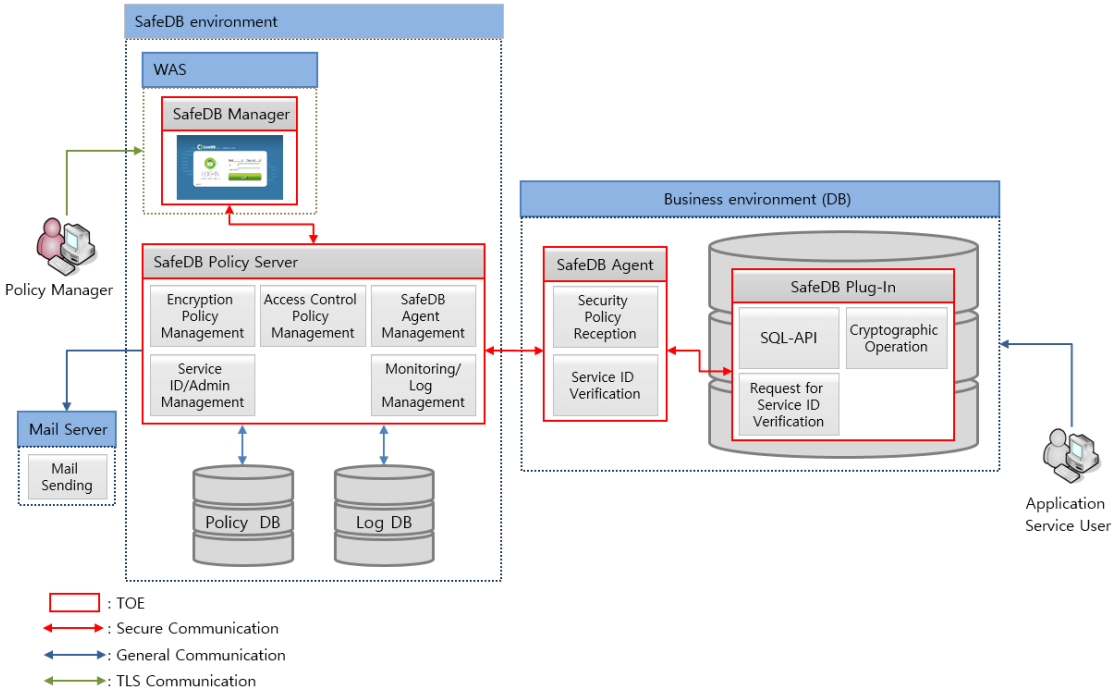
The TOE is classified into the API type and the plug-in type, depending on how it is used.

The API type encryption uses SafeDB SDK in the form of API on a business application and is irrelevant to an actual DB type, table name and column name, as it directly processes data encryption and decryption. SDK encryption and API encryption have the same meaning.



(Figure 1) TOE operational environment (API type)

The plug-in type is an encryption type that installs the SafeDB Plug-In module in the DBMS, and is used synonymously with the SQL-API type. In the case of the SQL-API type, an application service user inputs an encryption/decryption function in a query statement (SQL).



(Figure 2) TOE operational environment (Plug-In type)

1.3. TOE reference

[Table 2] TOE reference

TOE Identification	SafeDB V4.1		
TOE Version	V4.1 (version detail: V4.1.2)		
TOE Developer	INITECH Co., Ltd.		
TOE Component	Name	Version	Use
	SafeDB Policy Server	V4.1.2	Security policy management server
	SafeDB Manager	V4.1.2	Administrator UI
	SafeDB Agent	V4.1.2	Security policy deployment client
	SafeDB SDK for C	V4.1.2	Cryptographic operation
	SafeDB SDK for Java	V4.1.2	Cryptographic operation
	SafeDB Plug-In	V4.1.2	Cryptographic operation
Guidance Document	CCP.C_SB41_Preparative Procedure (PRE)_V1.3 CCP.C_SB41_Operational Guidance (OPE)_V1.2		
Delivery Date	June 07, 2024		

1.3.1. TOE components and major security features

The TOE is a DB encryption solution that can protect important data by providing the function of encrypting data stored in the DB, managing policies and keys used for encryption, and auditing the security setting history and set auditable events.

The TOE supports two types as follows:

1) SafeDB Plug-In type that can provide the DB security service only by installing it additionally in the DB without the need for modification or separate development process for the application
2) SafeDB SDK type that can provide the DB security service as API at the existing application level for the purpose of performance, DB load distribution and so forth

By providing both types at the same time, the TOE offers the DB security solution suitable for circumstances of an individual customer.

The TOE consists of SafeDB Policy Server that manages the encryption policy and keys; SafeDB Manager that performs the web security management; SafeDB Agent and SafeDB SDK installed and operated on the business (Application/DB) server; and SafeDB Plug-In applied to and operated in the DBMS.

[Table 3] TOE components

Classification	Description	
SafeDB Policy Server	Type	Software
	Function	[TSF] 1. Security policy management - Security policy generation/deletion - User data cryptographic key generation/deletion 2. SafeDB Agent management - SafeDB Agent information addition/deletion 3. Administrator and Service ID management - Administrator identification and authentication - Administrator information addition/deletion - Service ID addition/deletion 4. Security policy deployment

		- Automatic or manual deployment of security policies to SafeDB Agent 5. Monitoring/audit log management - Collection of Policy Server and Agent resources - Audit log generation/view [Non-TSF] 1. Access control policy management 2. Back up and restore TFS data encryption key
SafeDB Manager	Type	Software
	Function	[TSF] 1. Web-based security management - Registration/modification/deletion of administrator - Registration/modification/deletion of security policy - Registration/modification/deletion of SafeDB Agent - Registration/modification/deletion of Service ID - Audit log view
SafeDB Agent	Type	Software
	Function	[TSF] 1. Security policy reception - Generation of One Day Key upon initial start-up - Security policy reception and parsing - Storage of policy information encryption 2. Service ID verification - Verification of Service ID requested from SafeDB SDK 3. Audit log generation [Non-TSF] 1. Access control authority check
SafeDB SDK for C	Type	Software
	Function	[TSF] 1. Service ID verification request and policy reception - Request for verification of individual Service ID and security policy reception - Storage of user data cryptographic key encryption
SafeDB SDK for Java		

		2. Cryptographic operation - Access right check - Cryptographic operation processing 3. Audit log generation [Non-TFS] 1. Performs access control
SafeDB Plug-In	Type	Software
	Function	[TSF] 1. Service ID verification request and policy reception - Request for verification of individual Service ID and security policy reception - Storage of user data cryptographic key encryption 2. Cryptographic operation (SQL-API type) - Access right check - Cryptographic operation processing - When using query (SQL), SafeDB encryption/decryption function registered in the DB is used. 3. Audit log generation [Non-TFS] 1. Performs access control

The security policy and logs of the TOE are stored and managed in the DB. For the DB used for the security policy, Apache Derby is utilized, and policy information stored is encrypted to be stored. MariaDB is used for the log DB.

[Table 4] Policy/Log DB

Classification	Type	Description	Remarks
Policy DB	Apache Derby	File DB	V10.17.1.0
Log DB	MariaDB	RDBMS	V10.11.8

1.3.2. Required non-TOE operational environment

The TOE is a software-type product installed and operated on commercial hardware and an operating system platform. The requirements for non-TOE hardware/software that is an essential

element in operating the product but does not fall under the scope of the TOE are as follows.

[Table 5] Requirements for non-TOE operational environment

Category	Requirements for Non-TOE Operational Environment					
	SafeDB Policy Server	**SafeDB Manager**	**SafeDB Agent**	**SafeDB SDK for Java**	**SafeDB SDK for C**	**SafeDB Plug-In**
Operating System	Windows Server 2022 Std. x64					
CPU	Intel Core i7-1165G7 2.80GHz or higher					
Memory	16GB					
NIC	10/100/1000 Ethernet Port 1ea or more					
HDD	480 MB or higher necessary for installation of the TOE (Policy Server, Derby, Including MariaDB)	64MB or higher necessary for installation of the TOE (Including Tomcat)	33MB or higher necessary for installation of the TOE	8MB or higher necessary for installation of the TOE	4MB or higher necessary for installation of the TOE	5MB or higher necessary for installation of the TOE
Mandatory SW	Apache Derby v10.17.1.0 MariaDB v10.11.8 JDK 21 (21.0.3)	JDK 21 (21.0.3) Apache Tomcat v9.0.89	JDK 21 (21.0.3)	JDK 21 (21.0.3)	-	Oracle 19c

Mandatory software plays the following roles:

- JDK 21.0.3
  Runtime environment to execute an application developed with Java
- Apache Tomcat 9.0.89
  Web application server to start up SafeDB Manager
- Apache Derby v10.17.1.0
  Network server to access the DBMS that stores the security policy
- MariaDB v10.11.8
  DBMS that stores audit logs
- Oracle 19c

DBMS that stores user data encrypted through the SafeDB Plug-in product

[Table 6] Cryptographic module used in the TOE

Validated Cryptographic Module	KCMVP Information	Remarks
INISAFE Crypto for C V5.4	• Validation date: Jun. 19, 2023 • Validation number: CM-233-2028.6	- SafeDB Policy Server - SafeDB Manager - SafeDB Agent - SafeDB SDK for Java - SafeDB SDK for C - SafeDB Plug-In

[Table 7] IT operational environment for implementing security features of the TOE

Category	Description
Mail Server	3rd Party Mail Server A link is established to send an alarm email to the administrator if an administrator authentication fails, the repository of audit trail is full, SafeDB Agent abnormality occurs, or an event of a failed TOE self-test is detected. Mail Server supports general commercial mail servers.
DBMS	Policy DB, log DB Refer to [Table 4] Policy/Log DB

The requirements for administrator PC for the security management of the TOE are described in [Table 8] below:

[Table 8] Minimum requirements for administrator PC

Category	Minimum Requirements
Web browser	Microsoft Edge 120.0 or higher Google Chrome 120.0 or higher

1.4. TOE description

The TOE is software that encrypts important information in the DB and performs security functions, such as security audit, identification and authentication and security management.

The TOE consists of the following modules: SafeDB Policy Server that manages administrator and Service IDs, manages the security policy for encrypting the data of application service users, and provides the policy to SafeDB Agent; SafeDB Manager provided as a web-based interface (GUI) for the administrator; SafeDB Agent installed on each business server to receive the security policy and verify Service ID; SafeDB SDK in charge of actual cryptographic operation and log recording; and SafeDB Plug-In installed and operated (in charge of cryptographic operation and log recording) in the DBMS in the form of plug-in.

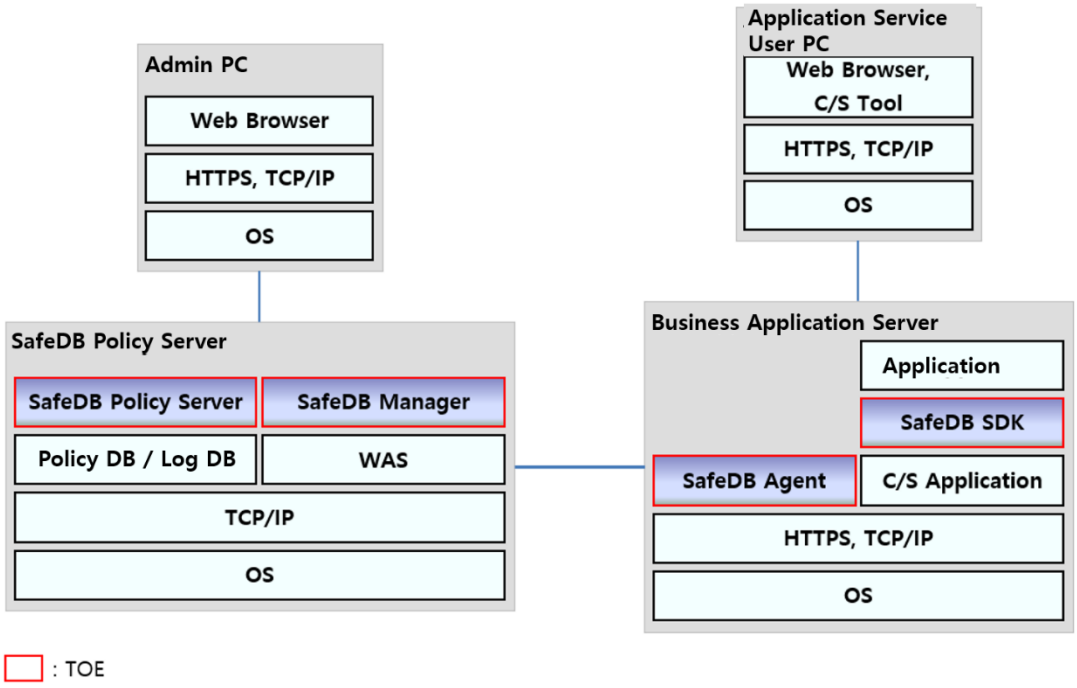
1.4.1. Physical scope of the TOE

The TOE components provided for consumers are classified into SafeDB Policy Server, SafeDB Manager, SafeDB Agent, SafeDB Plug-In, SafeDB SDK, the preparative procedure and the operational guidance as below:

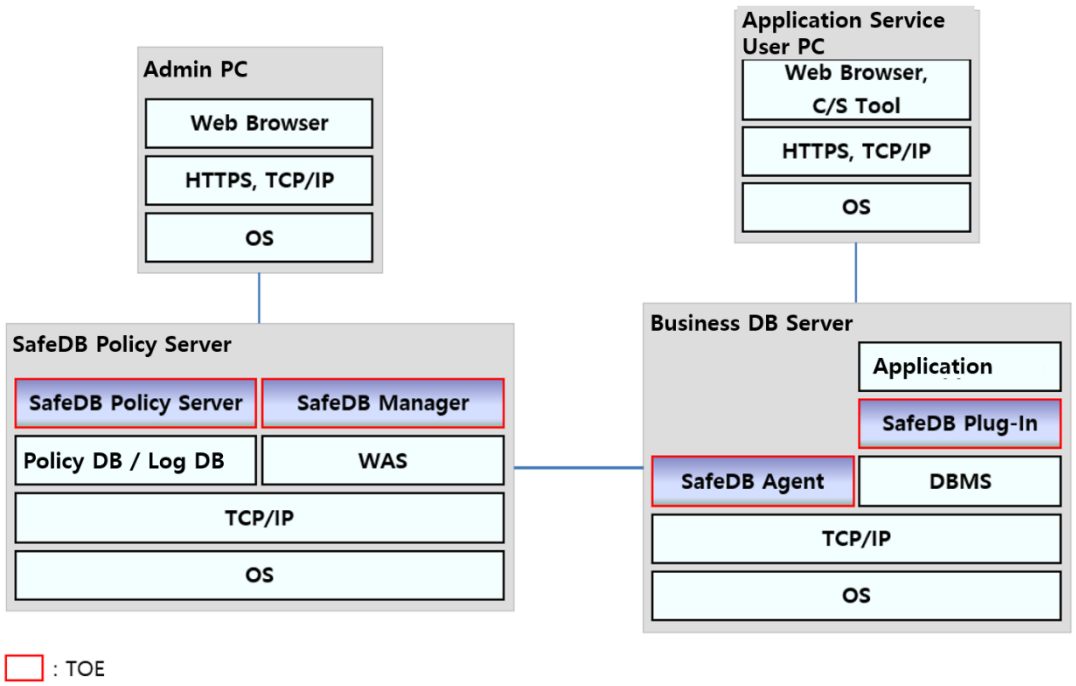
[Table 9] Physical scope of the TOE

Composition	Type	Identification Information			Within the TOE Scope
CD-ROM (1EA)	S/W	TOE installation program	SafeDB Policy Server	SafeDB_Server-4.1.2.zip	O
			SafeDB Manager	SafeDB_Manager-4.1.2.zip	O
			SafeDB Agent	SafeDB_Agent-4.1.2.zip	O
			SafeDB SDK for C	SafeDB_SDK_C-4.1.2.zip	O
			SafeDB SDK for Java	SafeDB_SDK_Java-4.1.2.zip	O
			SafeDB Plug-In	SafeDB_Plugin-4.1.2.zip	O
	Electronic document (PDF)	Guidance document	CCP.C_SB41_Preparative Procedure(PRE)_V1.3.pdf		O
			CCP.C_SB41_Operational Guidance(OPE)_V1.2.pdf		O
Certificate	Certificate	Software license certificate			X

The physical scope of the TOE includes SafeDB Policy Server, SafeDB Manager, SafeDB Agent, SafeDB Plug-In, and SafeDB SDK. The non-TOE operational environment is structured as below.



(Figure 3) Physical scope of the TOE (API type)



(Figure 4) Physical scope of the TOE (Plug-in type)

The physical scope of the TOE (API type) can be configured in two ways, as shown in Figure 3

and Figure 4. Physical scope of the TOE (API type).

TOE includes its own software, SafeDB Policy Server, SafeDB Manager, SafeDB Agent, SafeDB Plug-In, SafeDB SDK, Preparative Procedure, and Operational Guidance. The operator must prepare JDK, Apache Tomcat, Apache Derby, MariaDB, Oracle 19c, etc., which are essential software for TOE operation, before installation.

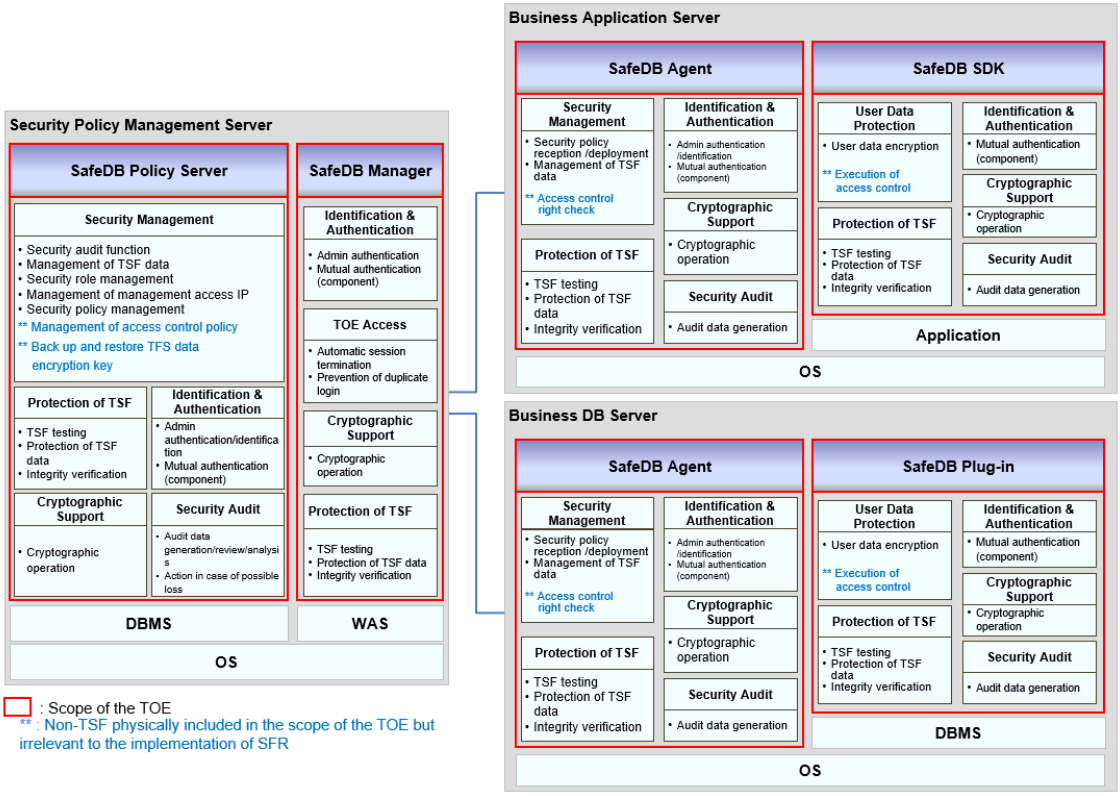
The non-TOE operational environment, such as hardware, operating system, web application server, database, java runtime environment, SSL environment, management console to control SafeDB Policy Server, and web browser to access SafeDB Manager, is excluded from the physical scope of the TOE.

The administrator's management access is provided via communication between a web browser on the administrator PC and the web server (Apache Tomcat), which is an operating environment of SafeDB Manager. TLS v1.2 encryption protocol is used to ensure a secure path.

1.4.2. TOE delivery method

The product package consists of CD 1EA and a certificate, as shown in the [Table 9] Physical scope of the TOE, and is delivered directly in person.

1.4.3. Logical scope of the TOE



(Figure 5) Logical scope of the TOE

The security functions provided by the TOE are as follows:

1) SafeDB Policy Server

- Security Management

  The security management function allows setting and management of security functions provided by the TOE, TSF data and so forth.

  - Security Function Management: It enables the administrator to manage security functions. The TOE provides the function of managing and monitoring security policies (encryption policy, agent information, administrator and Service ID information, etc.), and managing logs.

  - TSF Data Management: The TOE manages TSF data. For TSF data, it provides the function of managing security policies (encryption policy, SafeDB Agent information, administrator and Service ID information, etc.), managing passwords, viewing audit data, and so forth.

  - Security Role Management: The authorized roles of the TOE are classified into Security Manager (Policy Manager, General Manager) and Install Manager. Policy Manager can perform all web security management functions, while General Manager can perform

part of the web security management. Install Manager performs the security management through MasterConsole.

- Management Access IP Management: It performs the management of accessible IP of the administrator.

- Protection of the TSF
  The TOE performs TSF self-tests and conducts integrity verification on configuration files and TSF executable codes. It protects the stored TSF data, such as encryption key and administrator account, and data transmitted between SafeDB Policy Server and SafeDB Agent, and between SafeDB Policy Server and SafeDB Manager. In addition, it conducts tests on external entities (DBMS and mail server) interacting with the TOE.

- Cryptographic Support
  The TOE uses the KCMVP (Korea Cryptographic Module Validation Program)-validated cryptographic module to generate a cryptographic key for user data encryption and to distribute the cryptographic key. If the authorized administrator deletes an encryption policy, the TOE destroys the corresponding cryptographic key. The generated cryptographic key is encrypted by using the Master Key and stored in the Policy DB. In addition, it uses the KCMVP-validated cryptographic module and supports the cryptographic operation function, including symmetric key cryptography and hash performed for the TOE internal mutual authentication, the protection of the transmitted data, and the protection of the stored data.

- Identification and Authentication
  The TOE performs the identification and authentication in order to verify the administrator. If the identification and authentication attempts are unsuccessful for a defined number of times (fixed value of five times), the TOE performs the function of locking the account for a specified period of time (fixed value of 10 minutes), so that the TOE is protected against adverse attempts to authenticate a user. A locked administrator account can be unlocked by another authorized Policy Manager. In addition, the TOE verifies the combination rules and lengths of passwords that will be used for authentication, and does not provide feedback, including a reason for authentication failure during the authentication process. It also prevents the reuse of authentication data. The TOE performs mutual authentication between SafeDB Policy Server and TOE components (SafeDB Agent, SafeDB Manager).

- Security Audit
  Security Audit is a function to generate records of TOE use, such as the cryptographic operation function, and generates the audit data in a chronological order. The audit

records are stored in the Log DB, which is an operating environment of SafeDB Policy Server, and the authorized administrator can review the audit records. Additionally, if the audit data reach the threshold (fixed value of 90 percent), it is notified to the administrator via email. If an audit trail is full (fixed value of 95 percent), it ignores audited events and sends an email to the authorized administrator. Regarding the time used in audit records, timestamps are provided by the reliable operating system. In addition, the TOE analyzes potential security violations and takes actions to respond to a detected security violation.

2) SafeDB Manager

- Identification and Authentication

  The TOE performs the identification and authentication by sending to SafeDB Policy Server an administrator ID and password inputted from the Web Client (web browser) in order to verify the administrator. In the process of the identification and authentication, passwords being entered are masked (*) to prevent them from being disclosed. The TOE performs mutual authentication between SafeDB Manager and the TOE component of SafeDB Policy Server.

- TOE Access

  After a specified period of the administrator inactivity (fixed value of 10 minutes), the TOE terminates a session by making a request to WAS. In addition, it separately manages login information in order to prevent duplicate logins.

- Cryptographic Support

  The TOE supports cryptographic operation and cryptographic key management function performed for the TOE internal mutual authentication and the protection of the transmitted data by using the validated cryptographic module.

- Protection of the TSF

  The TOE performs TSF self-tests and conducts integrity verification. It provides basic protection of the stored TSF data and protects the data transmitted between SafeDB Manager and SafeDB Policy Server.

3) SafeDB Agent

- Security Management

  The security management function manages the security function provided by the TOE and the TSF data.

  - Security Function Management: Security Function Management: It performs the management of security functions. The TOE provides the function of receiving security policies assigned to SafeDB Agent and verifying service ID.

- TSF Data Management: The TOE manages the TSF data, such as cryptographic keys and security policies.

- Identification and Authentication
  It performs the identification and authentication of the Install Manager who manages the security function of SafeDB Agent. It also takes actions in response to failed authentication, protects authentication feedback and verifies the password combination rules. The TOE performs the mutual authentication between SafeDB Agent and the TOE components (SafeDB Policy Server, SafeDB SDK, SafeDB Plug-In).

- Protection of the TSF
  The TOE performs TSF self-tests and conducts integrity verification. It provides the protection of the stored TSF data (security policy) and the data transmitted between SafeDB Agent and the TOE components (SafeDB Policy Server, SafeDB SDK, SafeDB Plug-In).

- Cryptographic Support
  The TOE uses the validated cryptographic module and supports the cryptographic operation and cryptographic key management functions, such as symmetric key and hash performed for the TOE internal mutual authentication, the protection of the transmitted data and the protection of the stored data.

4) SafeDB SDK

- User Data Protection
  The TOE performs column-by-column encryption or decryption of user data by using cryptographic operations. In addition, the TOE ensures that any previous information content of a resource is made unavailable upon the resource allocation to and deallocation from user data.

- Protection of the TSF
  The TOE performs TSF self-tests and conducts integrity verification. It provides the protection of the stored TSF data and the data transmitted between SafeDB SDK and the TOE component (SafeDB Agent).

- Security Audit
  The TOE provides the function of generating audit data to check cryptographic operation results and so forth. Regarding the time used in audit records, timestamps are provided by the reliable operating system. Audit data generated in SafeDB SDK are transmitted to SafeDB Agent, and data transmitted among TOE components are protected by the function of the protection of the TSF.

- Cryptographic Support
  The TOE uses the validated cryptographic module and supports the cryptographic operation and cryptographic key management functions, such as symmetric key and hash performed for the TOE internal mutual authentication, the protection of the transmitted data and the encryption of user data and TSF data.

- Identification and Authentication
  The TOE performs the mutual authentication between SafeDB SDK and the TOE component (SafeDB Agent).

5) SafeDB Plug-In

SafeDB Plug-In has the same security functions as SafeDB SDK.

Non-TSF functions within the physical scope of the TOE but irrelevant to SFR implementation are as follows:

[SafeDB Policy Server]

- Access Control Policy Management: It provides the function for the authorized administrator to establish/manage relevant security policies, so that IP, MAC or time-based access control can be applied upon the user data encryption.
- Backup and Recovery Function: It provides the function of backing up and recovering a cryptographic key.

[SafeDB Agent, SafeDB SDK, SafeDB Plug-In]

- Access Control Function: It checks access control authority (performed in SafeDB Agent) and performs access control (SafeDB SDK, SafeDB Plug-In).

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.). For example, it is indicated as FAU_SAR.3(1) and FAU_SAR.3(2).

Assignment

This is used to assign specific values to unspecified parameters (e.g., password length). The result of assignment is indicated 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 *underlined and italicized.*

Refinement

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

1.6. Terms and definitions

The technical terms used in this ST are defined as follows. Terms used herein, which are the same as in the CC, must follow those in the CC.

Private Key

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

Object

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

Attack potential

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

Public key

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

Public key (asymmetric) cryptographic algorithm

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

Management access

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

Recommend/be recommended

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

Random bit generator (RBG)

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 an 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.

Symmetric cryptographic technique

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

Database (DB)

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.

Data Encryption Key (DEK)

Key that encrypts and decrypts the data

Iteration

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

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)

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

Protection Profile (PP)

Implementation-independent statement of security needs for a TOE type

Decryption

The act that restoring the ciphertext into the plaintext using the decryption key

Secret key

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

Session key

A cryptographic key that is used during one communication session between counterparties engaged in the communication. This is a temporary key and is used because if there are many ciphertexts using a single key, it is likely to analyze this to calculated the key

Certificate

Entity data that cannot be forged by using a private key or a secret key of public or private certificate authorities

Cryptographic module

A collection of hardware, software and/or firmware implementing a protection function subject to the validation (including cryptographic algorithm and key generation

Cryptographic boundary

A clearly defined continuing boundary that sets physical boundary of a cryptographic module. It includes components such as all hardware, software and/or firmware of a cryptographic module

User

Refer to "External entity"

It means an application service user, unless specified otherwise

User data

Data for the user, that does not affect the operation of the TSF (TOE security functionality)

Selection

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

Identity

Representation uniquely identifying an authorized user. The representation can be the full or abbreviated name or a pseudonym

Encryption

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

Element

Indivisible statement of a security need

Role

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

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 an object)

Specific type of action performed by a subject on an object

External entity

Entity (human or IT entity) interacting (or possibly interacting) with the TOE from outside of the TOE boundary

Threat agent

Unauthorized external entity that can pose illegitimate threats such as adverse access, modification or deletion to an asset

Authorized administrator

Authorized user who securely operates and manages the TOE

Authorized user

User who may, in accordance with the Safety Functional Requirements (SFR), perform an operation

Authentication data

Information used to verify the claimed identity of a user

Self-test

Pre-operational or conditional test executed by the cryptographic module

Assets

Entities that the owner of the TOE presumably places value upon

Refinement

Addition of details to a component

Organizational Security Policies

Set of security rules, procedures, or guidelines for an organization wherein the set is currently given by actual or virtual organizations, or is going to be given

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

Subject

Active entity in the TOE that performs operations on objects

Augmentation

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

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

Class

Set of CC families that share a common focus

Key Encryption Key (KEK)

Key that encrypts and decrypts another cryptographic key

Target of Evaluation (TOE)

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

Evaluation Assurance Level (EAL)

Set of assurance requirements drown 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

Assignment

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

Can/could

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

Shall/must

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

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)

Application Server

The application server defined in this PP 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.

Database Server

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

Database Management System (DBMS)

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

Secure Sockets Layer (SSL)

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

Transport Layer Security (TLS)

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 generated by the TOE and for the TOE, which can affect the operation of the TOE

Korea Cryptographic Module Validation Program (KCMVP)

Korea Cryptographic Module Validation Program

Security level

A combination of hierarchical Classification and non-hierarchical Category representing the importance of user or information

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

Policy Manager

An authorized user who securely operates and manages the TOE and has the entire privileges regarding the implementation of security management of the TOE

General Manager

An authorized user who securely operates and manages the TOE. General Manager manages the TOE within the scope of the view privileges granted by Policy Manager including security policy, resource management and monitoring, but except for settings and policy synchronization

Security Manager

It collectively refers to Policy Manager and General Manager. They are referred to as "administrator" unless otherwise specified

Install Manager

An administrator granted privileges to access the console program for the installation and operation of the TOE

Service ID

Identifier information to check the privilege to execute cryptographic operations of a TOE user (human or IT entity)

Application service user

A user who was granted a right to use through service ID verification and uses the user data encryption function by using the TOE. They are referred to as "user" unless otherwise specified

Application Programming Interface (API)

API means a function to use a certain system or be connected to a certain system. Generally, it is defined as a function that simplifies means and methods to access a provided system, which can be used by just calling it without knowing its internal structure.

Software Development Kit (SDK)

SDK used to mean a program development kit for Windows provided by Microsoft that develops programs using API, but changed to have the meaning same as API

MasterConsole

Program that performs the function of preliminary steps (Master Key initialization, policy initialization, encryption/decryption of configuration files, etc.), or the function of encryption and decryption of SafeDB Agent configuration files.

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.

Chapter 3 Security Objectives describes the security objectives of the TOE, the operational environment and the rationale of the security objectives.

Chapter 4 describes the definition of extended components.

Chapter 5 Security Requirements describes the product-based security functional requirements, the security assurance requirements, the security requirements and the rational of dependencies.

Chapter 6 provides the TOE summary specification.

2. Conformance Claim

This ST claims to conform to the followings:

2.1. CC conformance claim

Category	Conformance
Common Criteria	Common Criteria for Information Technology Security Evaluation V3.1R5 ● Common Criteria Part 1: Introduction and General Model V3.1r5, (CCMB-2017-04-001, 2017. 4) ● Common Criteria Part 2: Security Functional Components V3.1r5, (CCMB-2017-04-002, 2017. 4) ● Common Criteria Part 3: Security Assurance Components V3.1r5, (CCMB-2017-04-003, 2017. 4)
Part 2 Security Functional Requirements	Extended: FCS_RBG.1, FDP_UDE.1, FIA_IMA.1, FMT_PWD.1, FPT_PST.1, FTA_SSL.5
Part 3 Security Assurance Requirements	Conformant
Package	Augmented: EAL1 augmented (ATE_FUN.1)

2.2. PP conformance claim

This ST conforms to the "National Protection Profile for Database Encryption V1.1."

- PP title and version: National Protection Profile for Database Encryption
- Version: 1.1
- Certificate Number: KECS-PP-0820a-2017
- Certificate Date: Dec. 11, 2019
- Evaluation Assurance Level: EAL1+ (ATE_FUN.1)
- Conformance type: Strict PP conformance

2.3. Package conformance claim

This ST conforms to PP assurance requirement package EAL1 and additionally defines some assurance requirements.

- Assurance package: EAL1 augmented (ATE_FUN.1)

2.4. Conformance claim rationale

Since this ST adopts the TOE type, security objectives and security requirements in the same way as the Protection Profile, it is demonstrated that this ST strictly conforms to the "National Protection Profile for Database Encryption V1.1."

[Conformance Rationale]

- OE is augmented against the security objectives of the operational environment defined in the PP to which this ST conforms.
  - OE.SECURE_DBMS: augmented with conformance to PP selection SFR FAU_STG.1 requirement
  - OE.TIME_STAMP: augmented with conformance to PP selection SFR FPT_STM.1 requirement
  - OE.TRUSTED_PATH: augmented with conformance to PP selection SFR FTP_TRP.1 requirement

3. Security Objectives

This ST defines the security objectives for the operational environment only. The security objectives for the operational environment are those handled by IT area or non-technical/procedural methods.

3.1. Security objectives for the operational environment

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

[Table 10] Identification of security objectives for the operational environment

TOE Security Objective	Description
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 check 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_REINFORCEMENT	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.SECURE_DBMS	Security policies and audit records stored in the TOE are stored in the trusted database.

OE.TIME_STAMP	The TOE shall accurately record security-relevant events by using reliable time stamps provided by the TOE operational environment.
OE.TRUSTED_PATH	A secure path shall be ensured by the security policy of WAS when an authorized administrator accesses TOE administrator UI by using a web browser on PC.

4. Extended Components Definition

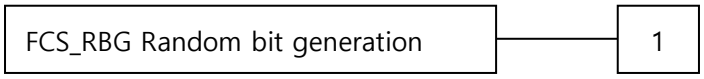
4.1. Cryptographic support (FCS)

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.

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*].

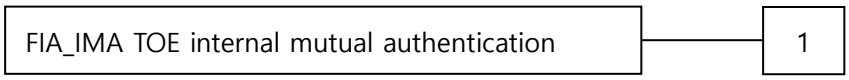
4.2. Identification & authentication (FIA)

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.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

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 the TOE*] by [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 (FDP_UDE, User Data Encryption) 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

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 methods*] specified.

4.4. Security management (FMT)

4.4.1. ID and password

Family Behaviour

This family (FMT_PWD, ID and password) 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 ID and password	1

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

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*] as follows:
1. [assignment: *password combination rules and/or length*]
2. [assignment: *other management such as management of special characters unusable for 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 password when installing, changing ID and password when the authorized administrator accesses for the first time*].

4.5. Protection of the TSF (FPT)

4.5.1. Protection of the stored TSF data

Family Behaviour

This family (FPT_PST, Protection of Stored TSF data) 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.

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*].

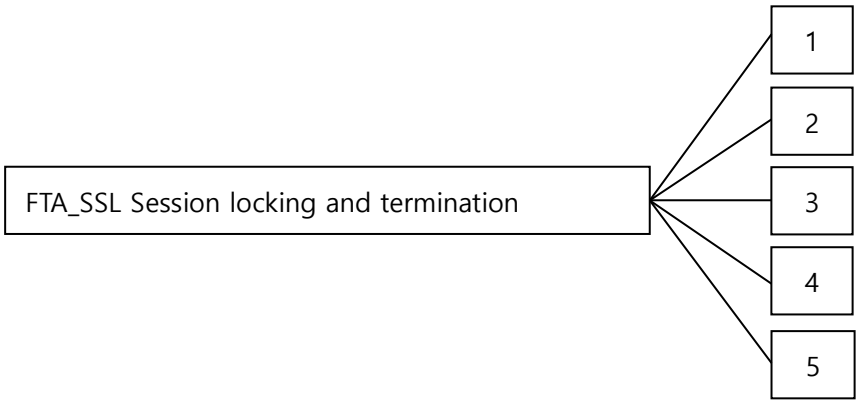
4.6. TOE access (FTA)

4.6.1. Session locking and termination

Family Behaviour

This family (FTA_SSL, Session locking and termination) defines requirement 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 the National Protection Profile for Database Encryption V1.1, 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 results in session locking or termination for each user
b) Specification of the default user inactivity period that results in session locking or 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

FTA_SSL.5 Management of TSF-initiated session
Hierarchical to No other components
Dependencies [FIA_UAU.1 Timing of authentication or no dependencies]

FTA_SSL.5.1 TSF shall [selection:
• *lock the session and/or 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.

5.1. Security functional requirements

All subjects, objects, operation, security attributes and external entities used in this ST are defined as follows.

For the subjects, objects, operations, security attributes, etc. used in the ST Security Requirements, refer to the corresponding SFRs. For external entities, refer to Chapter 1.

The security requirements in this ST consists of functional components in CC (CC V3.1_R5) Part 2. The following table summarizes the security functional components.

[Table 11] Summary of security functional components

Security Functional Class	Security Functional Component	
Security Audit (FAU)	FAU_ARP.1	Security alarms
	FAU_GEN.1	Audit data generation
	FAU_SAA.1	Potential violation analysis
	FAU_SAR.1	Audit review
	FAU_SAR.3	Selectable audit review
	FAU_STG.3	Action in case of possible audit data loss
	FAU_STG.4	Prevention of audit data loss
Cryptographic Support (FCS)	FCS_CKM.1(1)	Cryptographic key generation (user data encryption)
	FCS_CKM.1(2)	Cryptographic key generation (TSF data encryption)
	FCS_CKM.2	Cryptographic key distribution
	FCS_CKM.4	Cryptographic key destruction
	FCS_COP.1(1)	Cryptographic operation (user data encryption)
	FCS_COP.1(2)	Cryptographic operation (TSF data encryption)
	FCS_RBG.1(Extended)	Random bit generation
User Data Protection (FDP)	FDP_UDE.1(Extended)	User data encryption
	FDP_RIP.1	Subset residual information protection

Identification and Authentication (FIA)	FIA_AFL.1	Authentication failure handling
	FIA_IMA.1(Extended)	TOE internal mutual authentication
	FIA_SOS.1	Verification of secrets
	FIA_UAU.2	User authentication before any action
	FIA_UAU.4	Single-use authentication mechanisms
	FIA_UAU.7	Protected authentication feedback
	FIA_UID.2	User identification before any action
Security Management (FMT)	FMT_MOF.1	Management of security functions behaviour
	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
Protection of the TSF (FPT)	FPT_ITT.1	Basic internal TSF data transfer protection
	FPT_PST.1(Extended)	Basic protection of stored TSF data
	FPT_TEE.1	Testing of external entities
	FPT_TST.1	TSF testing
TOE Access (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

5.1.1. Security audit (FAU)

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 [*[Table 12] Actions against security violations*] upon detection of a potential security violation.

Regarding this requirement, SafeDB takes actions as follows:

[Table 12] Actions against security violations

Security Violation	Action
Accumulation of authentication failure events specified in FIA_UAU.2	· Limit login attempts for all authorized administrators for a specified period of time (fixed value of 10 minutes) · Send a warning email to the administrator

Integrity violation events and self-test failure events of the validated cryptographic module specified in FPT_TST.1	· Send a warning email to the administrator
Events of the audit trail disk capacity exceeding the limit, specified in FAU_STG.3	· Send a warning email to the administrator

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 13] Auditable events**, *[FPT_TEE.1 in [Table 13] Auditable events]*]

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, ["additional audit record" in **[Table 13]** Auditable events, *no other components*].

Regarding this requirement, SafeDB takes actions as follows.

[Table 13] Auditable events

Security Functional Component	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 Modification 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 modification in the behaviour 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	
FPT_TEE.1	Execution of tests of external entities and the results of the tests	

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) Accumulation or combination of [authentication failure audit event among auditable events of FIA_UAU.2, integrity violation audit event and self-test failure event of validated cryptographic module among auditable events of FPT_TST.1, [exceeding of audit storage threshold among auditable events of FAU_STG.3]] known to indicate a potential security violation

b) [Any other audit event rules including potential violation]

[

- FIA_UAU.2: if the authentication failure occurs repeatedly for five times among the administrator's authentication failure audit events
- FIA_STG.3: if the audit trail exceeds the limit of 90%

]

FAU_SAR.1 Audit review

Hierarchical to No other components

Dependencies FAU_GEN.1 Audit data generation

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

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

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 [the following methods of selection and/or ordering] of audit data based on [the following criteria with logical relations].

[

- Criterial with logical relations: Combination with AND operation when entering "selection criteria" value in [Table 14] Type of audit data and selection criteria
- Selection and/or ordering method: Sorting by descending order, based on audit data selection and the occurrence time of audit data in accordance with [Table 14] Type of audit data and selection criteria

]

[Table 14] Type of audit data and selection criteria

Audit Data Type	Selection Criteria	Allowed Capability
Plug-In Log (Plug-In service log)	Time and date of event (start date – end date)	Search, Sorting (based on the occurrence time)
	SID name	
	Service ID	
	Service ID's access IP	
	Service ID's Mac	
	Owner name	
	Table name	
	Column name	
	Log type	
	Log message	
SDK Log (SDK service log)	Time and date of event (start date – end date)	Search, Sorting (based on the occurrence time)
	SDK name	
	HOST name	
	Service ID's access IP	
	Service ID	
	IP of accessed SafeDB Agent: Port	
	Function (Login, Encrypt, Decrypt, Logout)	
	Log level (ERROR, DEBUG, INFO)	
	Table name	
	Column name	
	Policy name	
	Log message	
Manager Log (administrator log)	Time and date of event (start date – end date)	Search, Sorting (based on the occurrence time)
	Log type (all, common resource management, policy management)	
	Administrator ID	
	Administrator name	
	Administrator IP	
	Log summary	
Mail Log (Log of emails sent to	Time and date of event (start date – end date)	Search, Sorting (based on the

administrator)	Mail content	occurrence time)
Console Log (Log generated in MasterConsole)	Time and date of event	Search, Sorting (based on the occurrence time)
	Type	
	ID	
	Name	
	Console host (IP:Port)	
	Message	

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 shall [notification to the authorized administrator] if the audit trail exceeds [a percentage of available space out of the audit record storage capacity (fixed value of 90%)].

Application notes: The authorized administrator is notified via email. If audit trail fails to send audit logs from TOE modules of SafeDB SDK and SafeDB Plug-in to Log DB, the TSF sends audit data again to ensure that audit data are not lost.

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 shall [*ignore audited events*] and *[send an email to the authorized administrator*] if the audit trail is full.

5.1.2. Cryptographic support (FCS)

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 ["algorithm" of [Table 15] User data encryption algorithm and key sizes] and specified cryptographic key sizes ["cryptographic key size" of [Table 15] User data encryption algorithm and key

sizes] that meet the following ["list of standards" of [Table 15] User data encryption algorithm and key sizes].

[Table 15] User data encryption algorithm and key sizes

List of Standards	Encryption Method	Algorithm	Cryptographic Key Size	Usage
KS X 1213-1(2019) KS X 1213-2(2019) TTAK.KO-12.0271-Part1/R1(2016) TTAK.KO-12.0271-Part3(2017)	Symmetric key encryption	ARIA (CBC, CFB128, CTR, OFB)	128 192 256	User data encryption (symmetric key cryptographic operation)
KS X ISO/IEC 18033-3(2018) TTAS.KO-12.0004/R1(2005) TTAK.KO-12.0271-Part1/R1(2016) TTAK.KO-12.0274-Part4(2017)		SEED (CBC, CFB128, CTR, OFB)	128	
ISO/IEC 10118-3(2018)	Hash	SHA2	224 256 384 512	User data encryption (HASH)
KS X ISO/IEC 9797-2(2018) TTAK.KO-12.0330-Part2(2018)	Message authentication code	HMAC-SHA2	224 256 384 512	User data encryption (HMAC)
KS X ISO/IEC 18031(2018) TTAK.KO-12.0331-Part1(2018) TTAK.KO-12.0331-Part2(2018)	Random bit generator	HASH-DRBG-SHA2	256	For key generation

Application notes: The size of a key generated by the random bit generator is 32 bytes by default, and a key is generated to the size of an input factor (a multiple of 8, positive number). If a size according to the cryptographic algorithm is entered as a factor, a key of the corresponding size is generated.

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 ["algorithm" of [Table 16] Cryptographic key algorithm and key sizes] and specified cryptographic key sizes ["key size" of [Table 16] Cryptographic key algorithm and key sizes] that meet the following ["list of standards" of [Table 16] Cryptographic key algorithm and key sizes].

Algorithms and key sizes used for TSF data cryptographic key and cryptographic operation for secure communication are as follows:

[Table 16] Cryptographic key algorithm and key sizes for TSF data

List of Standards	Cryptographic Key Name	TOE Module	Algorithm	Key Size
ISO/IEC 18033-2(2016) IETF RFC 8017(2016)	Pubkey	SafeDB Policy Server	RSAES (SHA-256)	2048
	Privkey	SafeDB Policy Server		
	Inisafenet.cer	SafeDB Policy Server SafeDB Manager		
	Inisafenet.key	SafeDB Agent SafeDB SDK		
TTAK.KO-12.0334-Part1(2018) TTAK.KO-12.0334-Part2(2018)	driven key	SafeDB Policy Server SafeDB Agent	PBKDF2	256
KS X 1213-1(2019) KS X 1213-2(2019) TTAK.KO-12.0271-Part1/R1(2016) TTAK.KO-12.0271-Part3(2017)	master Key	SafeDB Policy Server	ARIA-CBC	256
	one-day Key	SafeDB Agent		
	dbpwd	SafeDB Policy Server		
	config key	SafeDB Policy Server SafeDB Agent SafeDB SDK SafeDB Plugin		
KS X ISO/IEC 18033-3(2018) TTAS.KO-12.0004/R1(2005)	Session Key	SafeDB Policy Server SafeDB Manager SafeDB Agent SafeDB SDK	SEED-CBC	128

TTAK.KO-12.0271-Part1/R1(2016) TTAK.KO-12.0274-Part4(2017)		SafeDB Plugin		
KS X ISO/IEC 9797-2(2018) TTAK.KO-12.0330-Part2(2018)	Integrity check	SafeDB Policy Server SafeDB Manager SafeDB Agent SafeDB SDK SafeDB Plugin	HMAC-SHA-2	256
ISO/IEC 10118-3(2018)	Administrator password hash	SafeDB Policy Server SafeDB Manager	SHA-2	-

Usage for each TSF data cryptographic is as follows:

[Table 17] Usage of TSF data cryptographic key

Category	Cryptographic Key Name	TOE Module	Usage
KEK	pubkey	SafeDB Policy Server	Protection of master key and dbpwd
	privkey	SafeDB Policy Server	Protection of master key and dbpwd
	inisafenet.cer	SafeDB Policy Server SafeDB Manager SafeDB Agent SafeDB SDK	Protection of session keys for encryption/decryption of data in transit in communication
	inisafenet.key		
	driven key	SafeDB Policy Server	Protection of masterkey, dbpwd, configkey
		SafeDB Agent	Protection of one-daykey, dbpwd, configkey
DEK	masterkey	SafeDB Policy Server	Encryption/decryption of security policy
	one-day key	SafeDB Agent	Encryption/decryption of user policy data
	dbpwd	SafeDB Policy Server	For encryption/decryption of account information for access to policy/log DB
	config key	SafeDB Policy Server SafeDB Agent SafeDB SDK SafeDB Plugin	Encryption/decryption of configuration file
	Session Key	SafeDB Policy Server SafeDB Manager SafeDB Agent SafeDB SDK	Encryption/decryption of data in transit in communication

		SafeDB Plugin	

Algorithms for each TSF data cryptographic key are as follows:

[Table 18] TSF data cryptographic key algorithms

Category	Cryptographic Key Name	Algorithm	Size	Generation Algorithm	Key Protection Method
KEK	pubkey	RSA	2048	HASH-DRBG-SHA256	-
	privkey				PKCS#8
	inisafenet.cer				-
	inisafenet.key				PKCS#8
	driven key	PBKDF2	256	HASH-DRBG-SHA256	-
DEK	masterkey	ARIA	256	HASH-DRBG-SHA256	Encryption using pubkey and induction key (PBKDF2) during KEK
	dbPwd				
	one-day key	ARIA	256	HASH-DRBG-SHA256	Encryption using DrivenKey(PBKDF2)
	Config key				
	Session Key	SEED	128	HASH-DRBG-SHA256	Encryption using the public key contained in the certificate (inisafenet.cer)

FCS_CKM.2 Cryptographic key distribution

Hierarchical to: No other components

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or **FCS_CKM.1(1) Cryptographic key generation** **FCS_CKM.1(2) 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 ["cryptographic key distribution method" of [Table 19] Cryptographic key distribution method] that meets the following ["list of standards" of [Table 19] Cryptographic key distribution method].

[Table 19] Cryptographic key distribution method

List of Standards	Distribution Target	Distribution Method
KS X ISO/IEC 11770-3:2008	DEK in FCS_CKM.1(1)	Secure transmission by encrypting communication with Handshake encryption method using a validated cryptographic module
KS X ISO/IEC 11770-3:2008	Session key for encryption of communication between TOE modules in FCS_CKM.1(2)	Handshake encryption method using a validated cryptographic module

FCS_CKM.4 Cryptographic key destruction

Hierarchical to: No other components

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or **FCS_CKM.1(1) Cryptographic key generation** **FCS_CKM.1(2) 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 20] Cryptographic key destruction method according to storage] that meets the following [list of standards].

[Table 20] Cryptographic key destruction method according to storage

Cryptographic Key Storage Location	Destruction Method	Destruction Method Details	Destruction Target	Timing of Destruction
DB	Deletion	Execute Delete SQL to delete in the DB	User data cryptographic key	When the administrator deletes the security policy
			Security policy information	
Memory	Memory zeroing	Overwrite the entire keys with "0"	Master Key / One-Day Key	When the process shuts down or calling

	Parameter initialization	Initialization by nullifying major parameters	Security policy list	log-out API
Memory	Memory release	Overwrite the entire keys with "0" and release the memory	Session Key	Immediately after terminating the communication
Memory	Memory zeroing or initialization	Overwrite the entire keys with "0" or initialize by nullifying	User data cryptographic key	Immediately after cryptographic operation

FCS_COP.1(1) Cryptographic operation (user data encryption)

Hierarchical to No other components

Dependencies [FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or **FCS_CKM.1(1) Cryptographic key generation**] FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1 The TSF shall perform ["encryption method" of [Table 15] User data encryption algorithm and key sizes] in accordance with a specified cryptographic algorithm ["algorithm" of [Table 15] User data encryption algorithm and key sizes] and cryptographic key sizes ["cryptographic key sizes" of [Table 15] User data encryption algorithm and key sizes] that meet the following ["list of standards" of [Table 15] User data encryption algorithm and key sizes].

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

Hierarchical to No other components

Dependencies [FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or **FCS_CKM.1(2) Cryptographic key generation**] FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1 The TSF shall perform ["encryption method" of [Table 16] Cryptographic key algorithm and key sizes] in accordance with a specified cryptographic algorithm ["algorithm" of [Table 16] Cryptographic key algorithm and key sizes] and cryptographic key sizes ["cryptographic key sizes" of [Table 16] Cryptographic key algorithm and key sizes] that meet the following ["list of standards" of [Table 16] Cryptographic key algorithm and key sizes].

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 with the validated cryptographic module using a specified random bit generator that meets the following [list of standards of [Table 21] Random bit generator].

[Table 21] Random bit generator

List of Standards	Random Bit Generator	Basis Function
KS X ISO/IEC 18031(2018) TTAK.KO-12.0331-Part1(2018) TTAK.KO-12.0331-Part2(2018)	HASH-DRBG-SHA256	HASH function

5.1.3. User data protection (FDP)

FDP_UDE.1 User data encryption (Extended)

Hierarchical to No other components

Dependencies **FCS_COP.1(1) Cryptographic operation**

FDP_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 by column, [none]].

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)

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 [the administrator authentication attempt].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been *met*, the TSF shall perform [the following list of actions].

[

- Inactivate the identification and authentication function (fixed value of 10 minutes)
- Send a warning email to the administrator

]

Application notes: If the number of the unsuccessful administrator authentication attempts exceeds five times, the identification and authentication function is inactivated, and normal login can be performed after 10 minutes. However, Policy Manager can unlock an account of another administrator whose account has been locked.

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 ["handshake encryption" that is an authentication protocol using the validated cryptographic module] in accordance with ["TOE internal mutual authentication section" of [Table 22] TOE internal mutual authentication].

[Table 22] TOE internal mutual authentication

TOE Internal Mutual Authentication Section		Validated Cryptographic Module
SafeDB Policy Server	SafeDB Manager	Refer to [Table 6] Cryptographic module used in the TOE
SafeDB Policy Server	SafeDB Agent	
SafeDB Agent	SafeDB SDK	
SafeDB Agent	SafeDB Plug-In	

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 quality metric].

[

a) Allowable characters

- Alphabet upper/lowercase letters (52 letters: a - z, A - Z), numbers (10 letters: 0 – 9), special characters that can be entered by using a keyboard (29 letters: ~,`,!,@,#,$,%,^,&,*,(,),-,_,=,+,₩,|,[,{,],},;,,,<,.,>,/,?)

b) Min/max length

- 9 – 16 digits

c) Combination rules

- A combination of three or more of the followings: English alphabets, numbers, and special characters
- Excluded if a password has three or more consecutive letters (numbers) that are adjacent
- Excluded if the same letter is entered three times or more
- Excluded if a password was the same as any of the last five passwords
- Three special characters on the keyboard (', ", ;) are excluded

d) Change frequency (the period during which the password is used)

- 0 – 999 days (the frequency is defined by the authorized administrator)

]

FIA_UAU.2 User authentication before any action

Hierarchical to FIA_UAU.1 Timing of authentication

Dependencies FIA_UID.1 Timing of identification

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

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].

[

- The TOE authenticates the administrator through password-based cryptographic authentication.
- The TOE stores and checks the time of authentication and the number of unsuccessful attempts in the DB upon every administrator authentication.
- The TOE stores authentication data in the memory upon every

administrator authentication to check duplicated authentication requests.

- The TOE stores and checks unique session ID in the memory upon password-based cryptographic authentication in order to prevent reuse of authentication data.

]

FIA_UAU.7 Protected authentication feedback

Hierarchical to No other components

Dependencies FIA_UAU.1 Timing of authentication

FIA_UAU.7.1 The TSF shall provide only ['*' character, entered characters not disclosed] while the authentication is in progress.

FIA_UID.2 User identification before any action

Hierarchical to FIA_UID.1 Timing of identification

Dependencies No dependencies

FIA_UID.2.1 The TSF shall successfully identify each user before allowing any other TSF-mediated actions on behalf of that user.

5.1.5. Security management (FMT)

FMT_MOF.1 Management of security functions behaviour

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 ***conduct management actions of*** the functions [[Table 23] List of the administrator's security functions] to [the authorized roles].

[Table 23] List of the administrator's security functions

Administrator Type	Category	Security Function	Role			
			Determine	Disable	Enable	Modify
General Manager	Resource management	Service ID and group management	O	-	-	-
		Common DB	O	-	-	-

		management				
		Shared-key management	O	-	-	-
		SafeDB Agent management	O	-	-	-
	Policy management	Encryption policy management	O	-	-	-
Policy Manager	Resource management	Service ID and group management	O	-	-	O
		Common DB management	O	-	-	O
		Shared-key management	O	-	-	-
		SafeDB Agent management	O	-	-	O
		Configuration management	O	-	-	-
	Policy management	Encryption policy management	O	O	O	O
		Security policy deployment	O	-	-	-
General Manager Policy Manager	Additional management	Cryptographic operation status	O	-	-	-
		Audit data review	O	-	-	-
Install Manager	Initialization	TOE initialization setting	O	-	O	-
	Integrity verification	Generation of configuration integrity verification file	O	-	O	-

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*** [[Table 24] TSF data list and

management items] to [the authorized roles].

[Table 24] TSF data list and management items

Administrator Type	TSF Data	Management Item				
		Query	Modify	Delete	Generate	Initialize
General Manager	Audit data	O	-	-	-	-
	Common resource	O	-	-	-	-
	Encryption policy	O	-	-	-	-
	Agent information	O	-	-	-	-
	Version information	O	-	-	-	-
Policy Manager	Audit data	O	-	-	-	-
	Common resource	O	O	O	O	-
	Encryption policy	O	O	O	O	-
	Agent information	O	O	O	O	-
	Admin account information	O	O	O	O	-
	Version information	O	-	-	-	-
Install Manager	Security policy	-	-	-	-	O
	Cryptographic key (Master Key, account information encryption key, etc.)	-	-	-	O	-
	Set value	-	O	-	-	-

Application notes: Install Manager shall initialize security policies and audit data to start up and operate SafeDB Policy Server, and generate cryptographic keys, such as Master Key.

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

management function, Service ID management function] to [none].

1. [none]
2. [none]

FMT_PWD.1.2 The TSF shall restrict the ability to manage the ID of [administrator management function, Service ID management function] to [none].

1. [none]
2. [none]

FMT_PWD.1.3 The TSF shall provide the capability for ***changing password when the authorized administrator accesses SafeDB Manager for the first time, setting ID and password of the authorized administrator of MasterConsole when installing.***

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: [the following list of management functions to be provided by the TSF]

[

- TSF function management: management functions specified in FMT_MOF.1
- TSF data management: management functions specified in FMT_MTD.1
- ID and password management: management functions specified in FMT_PWD.1

]

FMT_SMR.1 Security roles

Hierarchical to No other components

Dependencies FIA_UID.1 Timing of identification

FMT_SMR.1.1 The TSF shall maintain the roles [[Table 25] Classification and roles of administrators].

[Table 25] Classification and roles of administrators

Classification	Level	Roles
Security Manager	Policy Manager	Perform web security management functions

		(query, generate, modify and delete security policies)
	General Manager	Perform web security management functions (query security policies)
Install Manager	N/A	Perform installation security management functions

FMT_SMR.1.2 The SF shall be able to associate users and their roles **defined in FMT_SMR.1.1.**

5.1.6. Protection of the TSF (FPT)

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.

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] stored in containers controlled by the TSF from the unauthorized *disclosure, modification.*
[TSF data:

- Security policy: DB name, owner name, table name, column name, user data encryption key (DEK), encryption option
- Agent information, common resource information
- Administrator account information
- Policy DB access account information
- Master Key, One-Day Key, etc.
- TOE set values, such as the number of authentication failures, integrity test interval, IP/PORT information, etc.

]

[Table 26] Protection method in storing cryptographic keys and critical security parameters

Module	Encryption Target	Algorithm and Operation	Key Used	Storage Location	Application Method

		Method			
SafeDB Policy Server	Master Key, iv	ARIA/256/CBC	Password-based derivation key	File	1st encryption
	Master Key	RSA 2048	Private certificate public key	File	2nd encryption
SafeDB Agent	One-Day Key, iv	ARIA/256/CBC	Password-based derivation key	Memory	Encryption
	User Data Encryption Key (DEK)	ARIA/256/CBC	One-Day Key	Memory	Encryption

[Table 27] Security policy and account information encryption list

Category	Cryptographic Key	Encryption Method	Algorithm	Encryption List	Data Storage Location
SafeDB Policy Server	Master Key	Symmetric key encryption	ARIA (CBC)	All security policy information	Policy DB
SafeDB Policy Server	-	Hash	SHA-256	Administrator and Service ID and password	Policy DB
SafeDB Policy Server	dbpwd	Symmetric key encryption	ARIA (CBC)	Policy DB account information cipher	File
SafeDB Agent	One-Day Key	Symmetric key encryption	ARIA (CBC)	All security policy information	Memory

[Table 28] Configuration file encryption and integrity check algorithm and key

Category	Type	Algorithm	List of Standards	Generation Method
SafeDB Policy Server	Configuration file encryption	ARIA(CBC)	KS X 1213-1(2019) KS X 1213-2(2019)	Key generation using HASH-DRBG-SHA256

			TTAK.KO-12.0271-Part1/R1(2016) TTAK.KO-12.0271-Part3(2017)	
	Integrity check	HMAC-SHA-2	KS X ISO/IEC 9797-2(2018) TTAK.KO-12.0330-Part2(2018)	
SafeDB Agent	Configuration file encryption	ARIA(CBC)	KS X 1213-1(2019) KS X 1213-2(2019) TTAK.KO-12.0271-Part1/R1(2016) TTAK.KO-12.0271-Part3(2017)	Key generation using HASH-DRBG-SHA256
	Integrity check	HMAC-SHA-2	KS X ISO/IEC 9797-2(2018) TTAK.KO-12.0330-Part2(2018)	
SafeDB Manager	Integrity check	HMAC-SHA-2	KS X ISO/IEC 9797-2(2018) TTAK.KO-12.0330-Part2(2018)	Key generation using HASH-DRBG-SHA256
SafeDB SDK	Integrity check	HMAC-SHA-2	KS X ISO/IEC 9797-2(2018) TTAK.KO-12.0330-Part2(2018)	Key generation using HASH-DRBG-SHA256

SafeDB Plug-in	Integrity check	HMAC-SHA-2	KS X ISO/IEC 9797-2(2018) TTAK.KO-12.0330-Part2(2018)	Key generation using HASH-DRBG-SHA256

[Table 29] Storage for configuration file encryption Key and integrity check file

Category		Algorithm	Cryptographic Key	Application Method	Storage Location
SafeDB Policy Server	Configuration file encryption key	PBKDF2	configkey	Encryption	File
	Integrity check file	HMAC-SHA-2	-	HMAC	File
SafeDB Agent	Configuration file encryption key	PBKDF2	configkey	Encryption	File
	Integrity check file	HMAC-SHA-2	-	HMAC	File
SafeDB SDK	Integrity check file	HMAC-SHA-2	-	HMAC	File
SafeDB Plug-in	Integrity check file	HMAC-SHA-2	-	HMAC	File

FPT_TEE.1 Testing of external entities

Hierarchical to No other components

Dependencies No dependencies

FPT_TEE.1.1 The TSF shall run a suite of tests *during initial start-up* to check the fulfillment of [normal operation state of "List" in [Table 30] Testing of external entities].

[Table 30] Testing of external entities

List	Test Content	Action Responding to Failure
DBMS	Attempt to access the established Policy DB and Log DB	Print out errors and stop start-up

Mail Server	Attempts to access the established Mail Server	Print out error messages

FPT_TEE.1.2 If the test fails, the TSF shall [print out errors and stop start-up if DBMS tests fail, print out error messages if Mail Server tests fail].

FPT_TST.1 TSF testing

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 31] TOE self-test targets*].

[Table 31] TOE self-test targets

Category	Item	Content (Role)
SafeDB Policy Server	Cryptographic module	Self-tests
	Process	Determine if the start-up was successful during the initial start-up, and generate audit logs
SafeDB Agent	Cryptographic module	Self-tests
	Process	Determine if the start-up was successful during the initial start-up, and generate audit logs. Periodically check normal operation of the Agent, and transmit the status to Policy Server
SafeDB SDK	Cryptographic module	Self-tests
SafeDB Plug-In	Cryptographic module	Self-tests
SafeDB Manager	Cryptographic module	Self-tests

Application notes: during normal operation, the frequency of periodic self-tests is 60 minutes.

[Table 32] TOE integrity test targets

Category	Item	Content (Role)
SafeDB Policy Server	Environment configuration file	TOE environment configuration file
	Stored TSF execution code	TOE process
SafeDB Manager	All files	Files composing the TOE

SafeDB Agent	Environment configuration file	TOE environment configuration file
	Stored TSF execution code	TOE process
SafeDB SDK	Environment configuration file	TOE environment configuration file
	Stored TSF execution code	TOE execution module
SafeDB Plug-In	Environment configuration file	TOE environment configuration file
	Stored TSF execution code	TOE execution module

Application notes: during normal operation, the frequency of periodic integrity test is 60 minutes.

FPT_TST.1.2 The TSF shall provide **the authorized administrator** with the capability to verify the integrity of *[environment configuration file in "Item" of [Table 32] TOE integrity test target].*

FPT_TST.1.3 The TSF shall provide **the authorized administrator** with the capability to verify the integrity *of [stored TSF execution code in "Item" of [Table 32] TOE integrity test target]*.

5.1.7. TOE access (FTA)

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_SMF.1.1]

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.

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

Hierarchical to No other components
Dependencies No dependencies

FTA_SSL.5.1 The TSF shall *terminate the interactive session* after [10 minutes of the **administrator** inactivity].

FTA_TSE.1 TOE session establishment
Hierarchical to No other components
Dependencies No dependencies

FTA_TSE.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 having the same rights*].

Application notes: The number of connection IP provided by the TOE is set as two by default.

5.2. Security assurance requirement

Assurance requirements of this ST are composed of assurance components in Common Criteria (CC V3.1_R5) Part 3, and the evaluation assurance level is EAL1+ (ATE_FUN.1). The table below summarizes assurance components.

[Table 33] Summary of assurance components

Assurance Class	Assurance Component	
Security Target Evaluation	ASE_INT.1	ST introduction
	ASE_CCL.1	Conformance claim
	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	AVA_VAN.1	Vulnerability survey

Assessment		

5.2.1. Security Target evaluation

ASE_INT.1 ST introduction

Dependencies No dependencies

Developer action elements

ASE_INT.1.1D 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 summarise 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.

ASE_CCL.1 Conformance claim

Dependencies ASE_INT.1 ST introduction
ASE_ECD.1 Extended components definition
ASE_REQ.1 Stated security requirements

Developer action elements

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

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

Content and presentation elements

ASE_CCL.1.1C The conformance claim shall contain a CC conformance claim 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.

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.

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.

ASE_REQ.1 Stated security requirements

Dependencies ASE_ECD.1 Extended components definition

Developer action elements

ASE_REQ.1.1D The evaluator shall confirm that no extended component can be clearly expressed using existing components.

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.

ASE_TSS.1 TOE summary specification

Dependencies ASE_INT.1 ST introduction
ASE_REQ.1 Stated security requirements
ADV_FSP.1 Basic functional specification

Developer action elements

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

Content and presentation elements

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

Evaluator action elements

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

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

5.2.2. Development

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

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 or operational error), their consequences and implications for maintaining secure operation.

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

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

Evaluator action elements

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

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

ALC_CMC.1 Labelling of the TOE

Dependencies ALC_CMS.1 TOE CM coverage

Developer action elements

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

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 followings: 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

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.

ATE_IND.1 Independent testing: conformance

Dependencies ADV_FSP.1 Basic functional specification
AGD_OPE.1 Operational user guidance
AGD_PRE.1 Preparative procedures

Developer action elements

ATE_IND.1.1D The developer shall provide the TOE 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

AVA_VAN.1 Vulnerability survey

Dependencies ADV_FSP.1 Basic functional specification
AGD_OPE.1 Operational user guidance
AGD_PRE.1 Preparative procedures

Developer action elements
AVA_VAN.1.1D The developer shall provide the TOE 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 preparation 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 processing Basic attack potential.

5.3. Security requirements rationale

5.3.1. Dependency rationale of security functional requirements

The security functional requirements used in this ST satisfy dependencies as shown in the table below, and there is no component that does not satisfy the dependency.

The following table shows the dependency of functional components.

[Table 34] Dependency of the TOE SFRs

No.	SFR	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_CKM2 or FCS_COP.1(1)	10 or 12
		FCS_CKM.4	11
		FCS_RBG.1	14
9	FCS_CKM.1(2)	FCS_CKM.2 or FCS_COP.1(2)	10 or 13
		FCS_CKM.4	11
		FCS_RBG.1	14
10	FCS_CKM.2	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1(1), FCS_CKM.1(2)	8, 9
		FCS_CKM.4	11
11	FCS_CKM.4	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1(1), FCS_CKM.1(2)	8, 9
12	FCS_COP.1(1)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1(1)	8
		FCS_CKM.4	11
13	FCS_COP.1(2)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1(2)	9
		FCS_CKM.4	11
14	FCS_RBG.1	-	-
15	FDP_UDE.1	FCS_COP.1(1)	12
16	FDP_RIP.1	-	-
17	FIA_AFL.1	FIA_UAU.1	20
18	FIA_IMA.1	-	-
19	FIA_SOS.1	-	-
20	FIA_UAU.2	FIA_UID.1	23
21	FIA_UAU.4	-	-
22	FIA_UAU.7	FIA_UAU.1	20
23	FIA_UID.2	-	-
24	FMT_MOF.1	FMT_SMF.1	27
		FMT_SMR.1	28
25	FMT_MTD.1	FMT_SMF.1	27
		FMT_SMR.1	28
26	FMT_PWD.1	FMT_SMF.1	27

		FMT_SMR.1	28
27	FMT_SMF.1	-	-
28	FMT_SMR.1	FIA_UID.1	23
29	FPT_ITT.1	-	-
30	FPT_PST.1	-	-
31	FPT_TEE.1	-	-
32	FPT_TST.1	-	-
33	FTA_MCS.2	FIA_UID.1	23
34	FTA_SSL.5	-	-
35	FTA_TSE.1	-	-

Rationale (1): FAU_GEN.1 has the dependency on FPT_STM.1. However, reliable time stamps provided by the security objective OE.TIME_STAMP for the operational environment of this ST are used, thereby satisfying the dependency.

Rationale (2): FAU_STG.3 and FAU_STG.4 have a dependency on FAU_STG.1. However, it is protected from unauthorized deletion or modification by the security objective OE.SECURE_DBMS for the operational environment of this ST, thereby satisfying the dependency.

FIA_AFL.1, FIA_UAU.7 have FIA_UAU.1 as a dependency, but is satisfied by FIA_UAU.2 in a hierarchical relationship with FIA_UAU.1.

FIA_UAU.2, FMT_SMR.1, and FTA_MCS.2 have FIA_UID.1 as a subordinate relationship, but are satisfied by FIA_UAU.2 in hierarchical relationship with FIA_UID.1.

5.3.2. Dependency rationale of TOE assurance requirements

As the dependency of each assurance package provided in the CC is already satisfied, the rationale is omitted.

The augmented SAR ATE_FUN.1 has the dependency on ATE_COV.1. The rationale for the relevant dependency follows the National Protection Profile for Database Encryption V1.1 to which this ST is conformant.

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 the tests and the TSFIs.

6. TOE Summary Specification

This chapter provides brief and clear description of how the security functions are implemented in the TOE. It also describes how the SFRs are satisfied.

The following table is the list of the security functions specified in the TOE Summary Specification.

[Table 35] List of TOE security functions

Security Functional Class	Security Functional Component	
Security Audit (FAU)	FAU_ARP.1	Security alarms
	FAU_GEN.1	Audit data generation
	FAU_SAA.1	Potential violation analysis
	FAU_SAR.1	Audit review
	FAU_SAR.3	Selectable audit review
	FAU_STG.3	Action in case of possible audit data loss
	FAU_STG.4	Prevention of audit data loss
Cryptographic Support (FCS)	FCS_CKM.1(1)	Cryptographic key generation (user data encryption)
	FCS_CKM.1(2)	Cryptographic key generation (TSF data encryption)
	FCS_CKM.2	Cryptographic key distribution
	FCS_CKM.4	Cryptographic key destruction
	FCS_COP.1(1)	Cryptographic operation (user data encryption)
	FCS_COP.1(2)	Cryptographic operation (TSF data encryption)
	FCS_RBG.1(Extended)	Random bit generation
User Data Protection (FDP)	FDP_UDE.1(Extended)	User data encryption
	FDP_RIP.1	Subset residual information protection
Identification and Authentication (FIA)	FIA_AFL.1	Authentication failure handling
	FIA_IMA.1(Extended)	TOE internal mutual authentication
	FIA_SOS.1	Verification of secrets
	FIA_UAU.2	User authentication before any action
	FIA_UAU.4	Single-use authentication mechanisms
	FIA_UAU.7	Protected authentication feedback
	FIA_UID.2	User identification before any action
Security Management (FMT)	FMT_MOF.1	Management of security functions behaviour
	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
Protection of the TSF (FPT)	FPT_ITT.1	Basic internal TSF data transfer protection
	FPT_PST.1(Extended)	Basic protection of stored TSF data
	FPT_TEE.1	Testing of external entities
	FPT_TST.1	TSF testing
TOE Access (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

6.1. Security audit (FAU)

Security audit function of the TOE consists of security alarm (FAU_ARP.1) and audit data generation (FAU_GEN.1), among many functions of security audit (FAU).

6.1.1. Audit data generation

The following audit data are generated in the TOE, which are collected and stored by Log Daemon (satisfying FAU_GEN.1).

- Security log generated by the security management function
- Security log generated by the cryptographic operation function
- Security log, such as actions against potential security violation, identification and authentication, TSF self-tests, session termination, etc.

When audit data are generated for any change of TSF data value, the modified TSF data value is also included.

Each security log includes the following items to compose audit data:

- Time of audit occurrence
- Location in which the audit occurs
- Warning level: ERROR, DEBUG, INFO
- Audit message, etc.

SFR to be satisfied: FAU_GEN.1

6.1.2. Potential violation analysis and action

The TSF shall take actions described in [Table 36] Actions against security violations in case potential security violation is detected, based on audit records generated (refer to 6.1.1).

[Table 36] Actions against security violations

Security Violation	Action
Accumulation of authentication failures specified in FIA_UAU.2	- Limitation on login attempts for a specified period of time (fixed value of 10 minutes) for all authorized administrators - Sending a warning email to the administrator
Integrity violation event and failure of self-test that requires validation specified in FPT_TST.1	- Sending a warning email to the administrator
Audit trail exceeding certain pre-defined limits specified in FAU_STG.3	- Sending a warning email to the administrator

SFR to be satisfied: FAU_ARP.1, FAU_SAA.1

6.1.3. Management of audit storage

The TOE shall take the following actions if the audit data exceeds the storage limit (satisfying FAU_STG.3).

a) If the threshold pre-defined by the administrator is reached (90 percent), the administrator is notified via email.

b) If the audit trail is full (95 percent), an audited event is ignored in the audit data storage and the administrator is notified via email.

The percentage of the threshold represents how much (%) of the total capacity of HDD using log DB is in use. For example, if the threshold of 90 percent is set on a 100 GB HDD, it means that the threshold is reached when 90 GB is in use, with 10 GB available.

SFR to be satisfied: FAU_STG.3, FAU_STG.4

6.1.4. Audit data view and review

Audit data generated are stored in the audit storage (refer to 6.1.1), and the administrator views and reviews stored audit data through the screen interface (GUI) provided by SafeDB Manager (FAU_SAR.1).

- Service log
- Administrator log, etc.

The TSF provides the GUI function that enables the authorized administrator to read audit records after accessing SafeDB Manager.

For each type of audit data, when entering "selection criteria" value in [Table 37] Audit data types and selection criteria, the search result that has a combination with AND operation can be displayed.

[Table 37] Audit data types and selection criteria

Audit Data Type	Selection Criteria	Allowed Capability
Plug-In Log (Plug-In Service Log)	Event date and time (start date – end date)	Search, Sort (in the order of time of occurrence)
	SID name	
	Service ID	
	Access IP of service ID	
	Mac of service ID	
	Owner name	
	Table name	
	Column name	
	Log type	
	Log message	
SDK Log (SDK Service Log)	Event date and time (start date – end date)	Search, Sort (in the order of time of occurrence)
	SDK name	
	HOST name	
	Access IP of service ID	
	Service ID	
	IP of accessed SafeDB Agent: Port	
	Function (Login, Encrypt, Decrypt, Logout)	
	Log level (ERROR, DEBUG, INFO)	
	Table name	
	Column name	
	Policy name	
	Log message	
Manager Log (Administrator Log)	Event date and time (start date – end date)	Search, Sort (in the order of time of occurrence)
	Log type (all, common resource management, policy management)	
	Administrator ID	
	Administrator name	
	Administrator IP	
	Log summary	
Mail Log	Event date and time (start date – end date)	Search,

(Log of Emails Sent to the Administrator)	Mail content	Sort (in the order of time of occurrence)
Console Log (Log Generated in MasterConsole)	Event date and time	Search, Sort (in the order of time of occurrence)
	Type	
	ID	
	Name	
	Console host (IP:Port)	
	Message	

[Table 38] TOE log types

Log Name	Description
Plug-In Log	Details of the result of service ID validation request that a user performed through SafeDB Plug-In, processing history including cryptographic operation
SDK Log	Details of the result of service ID validation request that a user performed through SafeDB SDK, processing history including cryptographic operation
Manager Log	History of security policies and settings registered/modified/deleted by the administrator through SafeDB Manager
Mail Log	History of sending emails to the registered administrators in SafeDB Policy Server
Console Log	History of changes in configuration file, encryption and operation upon the initial installation of SafeDB Policy Server and Agent Master Console

SFR to be satisfied: FAU_SAR.1, FAU_SAR.3

6.2. Cryptographic support (FCS)

Cryptographic support function of the TOE consists of the establishment of security policies by the authorized administrator, and generation and distribution of cryptographic keys through a secure random bit generator in accordance with the security policies. Cryptographic operation is processed in the user application or inside the DBMS through SafeDB SDK or SafeDB Plug-In. A cryptographic key stored in the memory is destroyed when the administrator calls TOE process shut-

down command or calls SafeDB SDK log-out API.

6.2.1. Cryptographic key generation and random bit generation

The TOE generates user data encryption keys as follows (satisfying FCS_CKM.1(1)).

[Table 39] User data encryption key generation method

List of Standards	Encryption Method	Algorithm	Usage
KS X ISO/IEC 18031(2018) TTAK.KO-12.0331-Part1(2018) TTAK.KO-12.0331-Part2(2018)	Random bit generator	HASH-DRBG-SHA256	For key generation

The TOE generates Master Key/One-Day Key to encrypt TSF data, such as user information, configuration, security policy and agent information in accordance with [Table 40] TSF data encryption key generation method below (satisfying FCS_CKM.1(2)).

[Table 40] TSF data encryption key generation method

List of Standards	Encryption Method	Algorithm	Usage
KS X ISO/IEC 18031(2018) TTAK.KO-12.0331-Part1(2018) TTAK.KO-12.0331-Part2(2018)	Random bit generator	HASH-DRBG-SHA256	Generation of Master Key, One-Day Key, and Session Key

The TOE uses HASH-DRBG-SHA256 random bit generator in generating encryption keys. The length of a key generated by the random bit generator is 32 bytes by default, and a key is generated to the length of an input factor (a multiple of 8, positive number). If a length according to the cryptographic algorithm is entered, a key of the corresponding length is generated (satisfying FSC_RBG.1 (Extended)).

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

Public keys and private keys used in the TOE are generated in the following manner:

[Table 41] RSA key generation algorithm

Category	Value	Content
Input	Size of modulus n |n|	Selection of a size of modulus subject to KCMVP validation | n | = 2048, 3072 bits
Output	Public key P_{ub} = (n, e) Private key P_{riv} = (n, d)	Refer to [Table 42] RSA key generation method

[Table 42] RSA key generation method

Order	Item	Details
1	Selection of sufficiently big, different prime number p, q with similar size	"Random number generator" in [Table 21] Random bit generator is used in generating p and q. p and q are obtained from the output of the random bit generator bigger than the size of | p |, | q |
2	Generation of secure prime number	Miller-Rabin tests are conducted for a sufficient number of times in order to ensure that p and q are prime numbers. If | n | = 2048 bits: conducted at least 56 times (error probability: 2^{-112}) If | n | = 3072 bits: conducted at least 64 times (error probability: 2^{-128}) The difference between p and q shall exceed $2^{\|n\|/2 - 100}$. If | n | = 2048 bits: the difference between p and q exceeds 2^{924} If | n | = 3072 bits: the difference between p and q exceeds 2^{1436}
3	Selection of encryption exponent e	e : random number that satisfies GCD(e, λ(n)) = 1 (e > 2^{16})
4	Selection of decryption exponent d d : the only prime number d that satisfies e · d ≡ 1(mod λ(n)) Here, λ(n) = LCM(p -1, q-1)	The size of d shall exceed |n|/2 bits. If | n | = 2048 bits: the size of d exceeds 1024 bits If | n | = 3072 bits: the size of d exceeds 1536 bits d shall not exceed λ(n).

SFR to be satisfied: FCS_CKM.1(2)

6.2.2. Cryptographic key distribution

The security policy (including a DB data encryption key generated by using a validate cryptographic module) is distributed from SafeDB Policy Server to SafeDB Agent before the identification and authentication of Service ID (satisfying FCS_CKM.2).

After the user identification and authentication, the security policy (including a DB data encryption key generated by using a validate cryptographic module) is distributed from SafeDB Agent to SafeDB SDK and SafeDB Plug-In (satisfying FCS_CKM.2).

[Table 43] Cryptographic key distribution method

List of Standards	Order	Origin	Destination	Distribution Target	Distribution Method
KS X ISO/IEC 11770-3:2008	1	SafeDB Policy Server	SafeDB Agent	DEK	Distribution through secure communication by Handshake encryption method using the validated cryptographic module
KS X ISO/IEC 11770-3:2008	2	SafeDB Agent	SafeDB SDK, SafeDB Plug-In	DEK	Distribution through secure communication by Handshake encryption method using the validated cryptographic module

In addition, the session key distribution method for TOE internal mutual authentication and protection of transmitted data is described below.

[Table 44] Cryptographic key distribution method

List of Standards	TOE Component	TOE Component	Distribution Target	Distribution Method
KS X ISO/IEC 11770-3:2008	SafeDB Manager	SafeDB Policy Server	Session key for encryption of communication between TOE modules	Handshake encryption method using the validated cryptographic module

KS X ISO/IEC 11770-3:2008	SafeDB Policy Server	SafeDB Agent	Session key for encryption of communication between TOE modules	Handshake encryption method using the validated cryptographic module
KS X ISO/IEC 11770-3:2008	SafeDB Agent	SafeDB SDK	Session key for encryption of communication between TOE modules	Handshake encryption method using the validated cryptographic module
KS X ISO/IEC 11770-3:2008	SafeDB Plug-In	SafeDB Agent	Session key for encryption of communication between TOE modules	Handshake encryption method using the validated cryptographic module

SFR to be satisfied: FCS_CKM.2

6.2.3. Cryptographic key destruction

Master Key stored in the memory is deleted when the administrator calls SafeDB Policy Server Process shut-down command (satisfying FCS_CKM.4).

SafeDB Agent destroys One-Day Key and the security policy information (including DEK) stored in the memory when the administrator calls process shut-down command (satisfying FCS_CKM.4).

SafeDB SDK and SafeDB Plug-In destroy the security policy information (including DEK) stored in the memory when calling the Log-out function (satisfying FCS_CKM.4).

When the administrator deletes the security policy in SafeDB Policy Server, Delete SQL is executed to destroy DEK stored in the DB (satisfying FCS_CKM.4).

A session key used for the communication between TOE components is released from the memory and destroyed when the communication is terminated (satisfying FCS_CKM.4).

KEK derived with password, etc. is not stored, and the key is generated upon cryptographic operation and destroyed immediately after it is used (satisfying FCS_CKM.4).

Cryptographic key destruction methods per storage are as follows:

[Table 45] Cryptographic key destruction method per storage

Cryptographic Key Storage	Destruction Method	Detailed Destruction Method	Destruction Target	Timing of Destruction
DB	Deletion	Execute Delete SQL to delete from the DB	User data encryption key	When the administrator deletes the security policy
			Security policy information	
Memory	Memory zeroing	Overwrite all the keys with "0"	Master Key One-Day Key	When calling process shut-down or log-out API
	Parameter initialization	initialization by nullifying major parameters	List of security policies	
Memory	Memory release	Overwrite all the keys with "0" and release the memory	Session Key	When terminating the communication
Memory	Memory zeroing or initialization	Overwrite all the keys with "0" or initialize by nullifying	User data encryption key	Immediately after cryptographic operation

SFR to be satisfied: FCS_CKM.4

6.2.4. Cryptographic operation

Cryptographic operation of the TOE is classified into the function of cryptographic operation for the prevention of disclosure and security of security policies, and the function of cryptographic operation of user DB data (satisfying FCS_COP.1(1), FCS_COP.1(2)).

Algorithms and key sizes used in cryptographic operation of user data are as follows

[Table 46] Cryptographic algorithms and key sizes for user data

List of Standards	Encryption Method	Algorithm	Cryptographic Key Size	Usage
KS X 1213-1(2019) KS X 1213-2(2019) TTAK.KO-12.0271-Part1/R1(2016) TTAK.KO-12.0271-	Symmetric key encryption	ARIA (CBC/CTR/CFB128/OFB)	128 192 256	User data encryption (symmetric key cryptographic operation)

Part3(2017)				
KS X ISO/IEC 18033-3(2018) TTAS.KO-12.0004/R1(2005) TTAK.KO-12.0271-Part1/R1(2016) TTAK.KO-12.0274-Part4(2017)	Symmetric key encryption	SEED (CBC/CTR/CFB128/OFB)	128	
ISO/IEC 10118-3(2018)	Hash	SHA2	224 256 384 512	User data encryption (HASH)
KS X ISO/IEC 9797-2(2018) TTAK.KO-12.0330-Part2(2018)	Message authentication code	HMAC-SHA2	224 256 384 512	User data encryption (HMAC)
KS X ISO/IEC 18031(2018) TTAK.KO-12.0331-Part1(2018) TTAK.KO-12.0331-Part2(2018)	Random bit generator	HASH-DRBG-SHA2	256	For key generation

Algorithms and key sizes used in cryptographic operation of TSF data cryptographic key and encrypted communication are as follows:

[Table 47] Cryptographic key algorithms and key sizes for TSF data

List of Standards	Cryptographic Key Name	TOE Module	Algorithm	Key Size
ISO/IEC 18033-2(2016) IETF RFC 8017(2016)	Pubkey	SafeDB Policy Server	RSAES (SHA-256)	2048
	Privkey			
	Inisafenet.cer	SafeDB Policy Server SafeDB Manager SafeDB Agent SafeDB SDK		
	Inisafenet.key			
TTAK.KO-12.0334-Part1(2018) TTAK.KO-12.0334-Part2(2018)	driven key	SafeDB Policy Server SafeDB Agent	PBKDF2	256
KS X 1213-1(2019)	Master Key	SafeDB Policy Server	ARIA-CBC	256

KS X 1213-2(2019) TTAK.KO-12.0271-Part1/R1(2016) TTAK.KO-12.0271-Part3(2017)	dbpwd	SafeDB Policy Server		
	One-Day Key	SafeDB Agent		
	Config key	SafeDB Policy Server SafeDB Agent SafeDB SDK SafeDB Plugin		
KS X ISO/IEC 18033-3(2018) TTAS.KO-12.0004/R1(2005) TTAK.KO-12.0271-Part1/R1(2016) TTAK.KO-12.0274-Part4(2017)	Session Key	SafeDB Policy Server SafeDB Manager SafeDB Agent SafeDB SDK SafeDB Plugin	SEED-CBC	128

Table 1

Usage of each cryptographic key for TSF data is as follows:

[Table 48] Usage of cryptographic key for TSF data

Category	Cryptographic Key Name	TOE Module	Usage
KEK	Pubkey	SafeDB Policy Server SafeDB Agent	Protection of master key and dbpwd
	Privkey	SafeDB Policy Server SafeDB Agent	Protection of master key and dbpwd
	INISAFENet.cer	SafeDB Policy Server SafeDB Manager SafeDB Agent SafeDB SDK	Protection of session keys for encryption/decryption of data in transit in communication
	INISAFENet.key		
	driven key	SafeDB Policy Server	Protection of masterkey, dbpwd, configkey
		SafeDB Agent	Protection of one-daykey, dbpwd, configkey
DEK	masterkey	SafeDB Policy Server	Encryption/decryption of security policy
	one-day key	SafeDB Agent	Encryption/decryption of user policy data
	dbpwd	SafeDB Policy Server	For encryption/decryption of account information for access to policy/log DB
	config key	SafeDB Policy Server SafeDB Agent SafeDB SDK SafeDB Plugin	Encryption/decryption of configuration file

	Session Key	SafeDB Policy Server SafeDB Manager SafeDB Agent SafeDB SDK SafeDB Plugin	Encryption/decryption of data in transit in communication

Information on algorithm for each cryptographic key for TSF data is as follows:

[Table 49] Information on cryptographic key algorithm for TSF data

Category	Cryptographic Key Name	Algorithm	Size	Generation Algorithm	Key Protection Method
KEK	Pubkey	RSA	2048	HASH-DRBG-SHA256	-
	Privkey				PKCS#8
	Inisafenet.cer				-
	Inisafenet.key				PKCS#8
	driven key	PBKDF2	256	HASH-DRBG-SHA256	-
DEK	masterkey	ARIA	256	HASH-DRBG-SHA256	Encryption using pubkey and induction key (PBKDF2) during KEK
	dbPwd				
	one-day key	ARIA	256	HASH-DRBG-SHA256	Encryption using DrivenKey(PBKDF2)
	masterkey				
	Session Key	SEED	128	HASH-DRBG-SHA256	Encryption using the public key contained in the certificate (inisafenet.cer)

SFR to be satisfied: FCS_COP.1(1), FCS_COP.1(2)

6.3. User data protection (FDP)

6.3.1. User data protection

The TOE provides the function of encryption/decryption by column in order to protect user data. It also ensures the security of residual information by executing memory initialization of major information when terminating the TOE process.

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

- API type

The authorized administrator establishes the security policy, such as cryptographic key generation, on SafeDB Policy Server, and then deploys a cryptographic key and the security policy to SafeDB Agent. SafeDB Agent receives the security policy, and then encrypts and stores it. When a request for Service ID validation is made in SafeDB SDK installed in the user application system, etc. to check the right to user data encryption, SafeDB Agent completes Service ID verification, and then deploys the security policy to SafeDB SDK. User data encryption/decryption is performed by calling an encryption interface in SafeDB SDK.

- Plug-In type

The authorized administrator establishes the security policy, such as cryptographic key generation, on SafeDB Policy Server, and then deploys the security policy to SafeDB Agent. SafeDB Agent receives the security policy, and then encrypts and stores it. When a request for Service ID verification is made in SafeDB Plug-In installed in the DBMS to check the right to user data encryption, SafeDB Agent completes Service ID verification, and then deploys the security policy to SafeDB Plug-In. User data encryption/decryption is performed by calling an encryption interface in SafeDB Plug-In.

SFR to be satisfied: FDP_UDE.1(Extended), FDP_RIP.1

6.4. Identification and authentication (FIA)

The identification and authentication function of the TOE consists of administrator identification and authentication for the purpose of performing the security management through SafeDB Manager, and TOE internal mutual authentication.

6.4.1. Administrator identification and authentication

Security Manager (Policy Manager and General Manager) can perform the web security management after undergoing the identification and authentication of SafeDB Manager. Install Manager can perform the console security management after undergoing the identification and authentication of Master Console in SafeDB Policy Server.

The administrator shall enter the following identification and authentication data through the screen interface (satisfying FIA_UID.2 and FIA_UAU.2)

- Administrator ID
- Administrator password

If the number of unsuccessful authentication attempts exceeds the limit (fixed value of five times), the account of the relevant ID is locked (fixed value of 10 minutes) to prevent repetitive attempts for the identification and authentication process (satisfying FIA_AFL.1).
The function of identification and authentication uses an authentication mechanism through password requiring a probabilistic analysis.

The TOE provides mechanisms to verify that administrator password meets the following defined quality metrics upon password registration or change (FIA_SOS.1).

a) Allowable characters
- Alphabet upper/lowercase letters (52 letters: a - z, A - Z), numbers (10 letters: 0 – 9), special characters that can be entered by using a keyboard (29 letters: ~,`,!,@,#,$,%,^,&,*,(,),-,_,=,+,₩,|,[,{,],},:,,,<,.,>,/,?)

b) Min/max length
- 9 – 16 digits

c) Combination rules
- A combination of three or more of the followings: English alphabets, numbers, and special characters
- Excluded if a password has three or more consecutive letters (numbers) that are adjacent
- Excluded if the same letter is entered three times or more
- Excluded if a password was the same as any of the last five passwords
- Three special characters on the keyboard (',",;) are excluded

d) Change frequency (the period during which the password is used)
- 0 – 999 days (the frequency is defined by the authorized administrator)

The TOE prevents reuse of authentication data as follows (FIA_UAU.4).
- The TOE authenticates the administrator through password-based cryptographic authentication.
- The TOE stores and checks the time of authentication and the number of unsuccessful attempts in the DB upon every administrator authentication.
- The TOE stores authentication data in the memory upon every administrator authentication to check duplicated authentication requests.
- The TOE stores and checks unique session ID in the memory upon password-based cryptographic authentication in order to prevent reuse of authentication data.

The TSF masks passwords with "*" in SafeDB Manager for users and does not display input characters in the administrator console (FIA_UAU.7)

SFR to be satisfied: FIA_AFL.1, FIA_SOS.1, FIA_UAU.2, FIA_UAU.4, FIA_UAU.7, FIA_UID.2

6.4.2. TOE internal mutual authentication

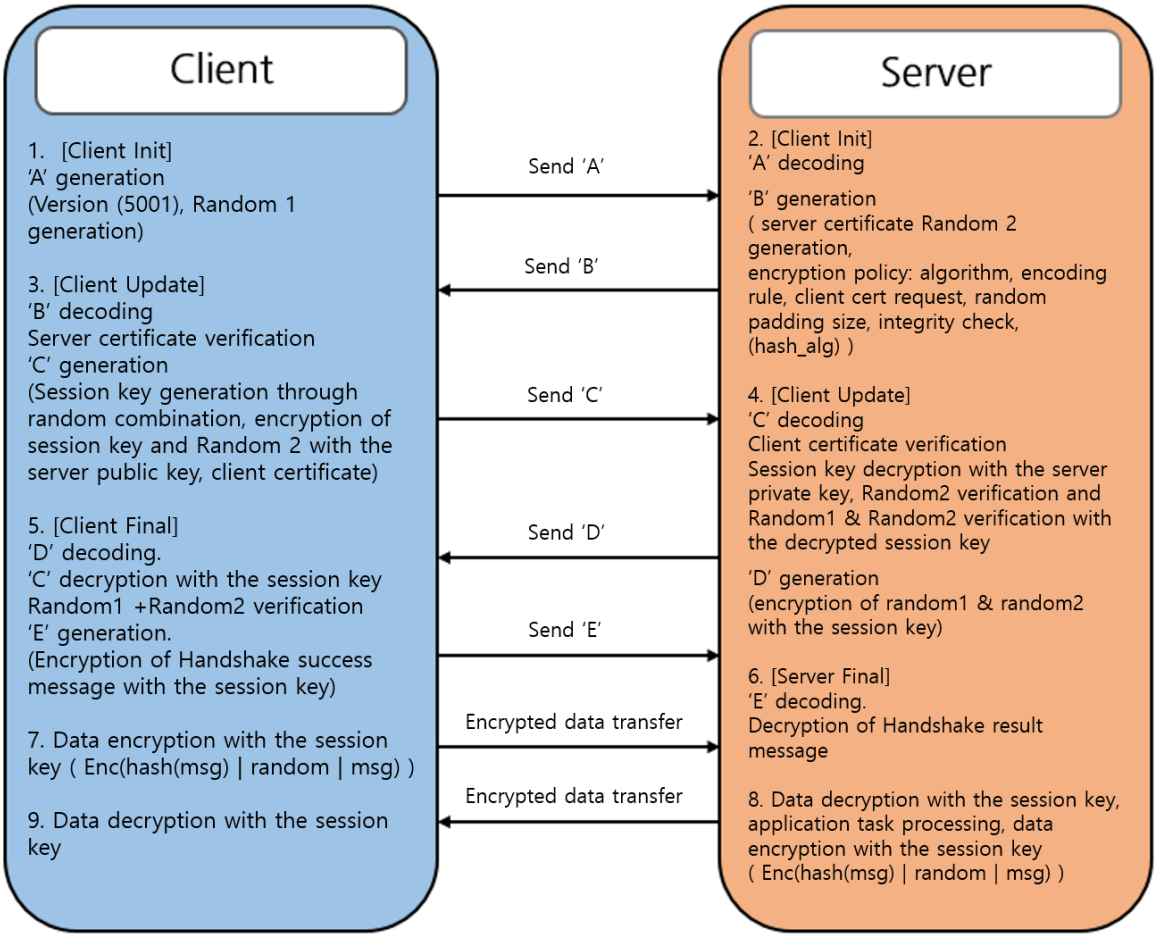
The TOE uses the security function in the Handshake encryption method using the validated cryptographic module for the purpose of secure TSF data transfer among TOE components in order to perform TOE internal mutual authentication. Then, it performs encrypted communication between components.

The following table describes TOE internal mutual authentication method and when the mutual authentication runs.

[Table 50] TOE internal mutual authentication

Origin (Client)	Destination (Server)	Mutual Authentication Method	When Mutual Authentication Runs
SafeDB Manager	SafeDB Policy Server	Handshake encryption method	When the administrator logs in to access SafeDB Manager
SafeDB Agent	SafeDB Policy Server	Handshake encryption method	When the Agent starts up
SafeDB SDK	SafeDB Agent	Handshake encryption method	Upon the request of user ID verification
SafeDB Plug-In	SafeDB Agent	Handshake encryption method	Upon the request of user ID verification
SafeDB Policy Server	SafeDB Agent	Handshake encryption method	When the function of Agent synchronization is performed by the administrator

TOE internal mutual authentication mechanism is as follows.



(Figure 6) Handshake encryption method

[Table 51] Handshake encrypted communication procedure

Step	Implementation	Behaviour
1	Client	Generate Random as specified in [Table 21] Random bit generator.
	Communication channel	Send Version + Random1 to the server. Send 'A' in [Handshake encryption method].
2	Server	Generate Random2 as specified in [Table 21] Random bit generator.
	Communication channel	The server sends the server certificate, Random2, and the encryption policy to the client. Send 'B' in [Handshake encryption method].
3	Client	Verify the validity of the server certificate.
	Client	Generate a session key as specified in [Table 21] Random bit generator.
	Client	Encrypt Random2 with the session key.

	Client	Encrypt the session key with a public key.
	Communication channel	Send encrypted Random2 + the session key encrypted with the public key + the client certificate to the server. Send 'C' in [Handshake encryption method].
4	Server	Verify the validity of the client certificate.
	Server	Decrypt the session key with a private key.
	Server	Decrypt Random2 with the decrypted session key.
	Server	Encrypt Random1 + Random2 with the session key.
	Communication channel	Send Random1+Random2 encrypted with a symmetric key. Send 'D' in [Handshake encryption method].
5	Client	Decrypt encrypted Random1+Random2 with the session key.
	Client	Verify Random1+Random2.
	Client	Encrypt the handshake result message with the session key.
	Communication channel	Send the encrypted result message. Send 'E' in [Handshake encryption method].
6	Server	Decrypt the encrypted result message with the session key.
7	Client/ Server	Encrypt the data that the user wants with the session key. The data encryption format is Enc(hash(msg) | random | msg). msg: data to be encrypted random: generate random in the defined bytes (HashDRBG) and do padding. To prevent the same cyphertext from being generated every time the same msg is encrypted. Hash(msg): Generate Hash value of msg through the set hash algorithm (Sha256). To verify the integrity of the message Enc: encrypt with the set symmetric key algorithm (ARIA-CBC, etc.)
	Communication channel	Send the data encrypted with the session key to the counterpart. Send the encrypted data in [Handshake encryption method]
8, 9	Server/Client	Decrypt the received encrypted message with the session key.

SFR to be satisfied: FIA_IMA.1(Extended)

6.5. Security management (FMT)

6.5.1. Management of security functions behaviour

The TOE provides the following function to manage security functions behaviours.

[Table 52] List of the administrator's security functions

Administrator	Category	Security Function	Role

Type			Determine	Disable	Enable	Modify
General Manager	Resource management	Service ID and group management	O	-	-	-
		Common DB management	O	-	-	-
		Shared-key management	O	-	-	-
		SafeDB Agent management	O	-	-	-
	Policy management	Encryption policy management	O	-	-	-
Policy Manager	Resource management	Service ID and group management	O	-	-	O
		Common DB management	O	-	-	O
		Shared-key management	O	-	-	-
		SafeDB Agent management	O	-	-	O
		Configuration management	O	-	-	-
	Policy management	Encryption policy management	O	O	O	O
		Security policy deployment	O	-	-	-
General Manager Policy Manager	Additional management	Cryptographic operation status	O	-	-	-
		Audit data review	O	-	-	-
Install Manager	Initialization	TOE initialization setting	O	-	O	-
	Integrity verification	Generation of configuration integrity verification file	O	-	O	-

Administrators are classified into Security Manager (Policy Manager, General Manager) and

Install Manager, and perform the security management, such as web security management and the security policy establishment according to their roles in MasterConsole.

SFR to be satisfied: FMT_MOF.1, FMT_SMF.1, FMT_SMR.1

6.5.2. Management of TSF data

The ability of the authorized administrator to manage TSF data is described below (FMT_MTD.1).

[Table 53] TSF data list and management roles

Administrator Type	TSF Data	Role				
		Query	Modify	Delete	Generate	Initialize
General Manager	Audit data	O	-	-	-	
	Common resource	O	-	-	-	
	Encryption policy	O	-	-	-	
	Agent information	O	-	-	-	
	Version information	O				
Policy Manager	Audit data	O	-	-	-	
	Common resource	O	O	O	O	
	Encryption policy	O	O	O	O	
	Agent information	O	O	O	O	
	Admin account information	O	O	O	O	
	Version information	O				
Install Manager	Security policy					O
	Cryptographic key (Master Key, account information encryption key, etc.)				O	
	Audit data					O
	Set value		O			

Application notes: Install Manager shall initialize security policies and audit data to start up and operate SafeDB Policy Server, and generate cryptographic keys, such as Master Key.

SFR to be satisfied: FMT_MOF.1, FMT_MTD.1, FMT_SMF.1, FMT_SMR.1

6.5.3. Management of security password

The TOE provides the function to register and modify user's authentication data – password. The TSF also provides the function that enables Policy Manager to change the password when he/she accesses SafeDB Manager for the first time in the installation process. Only the authorized Policy Manager can register and change a password through the menu provided by SafeDB Manager. In addition, it provides the function that enables Install Manager to set ID and password upon the initial access to MasterConsole of SafeDB Policy Server and Agent.
ID and password are encrypted with SHA-256 algorithm, and stored. Refer to FIA_SOS.1 for password lengths and combination rules selected when registering or changing a security password.

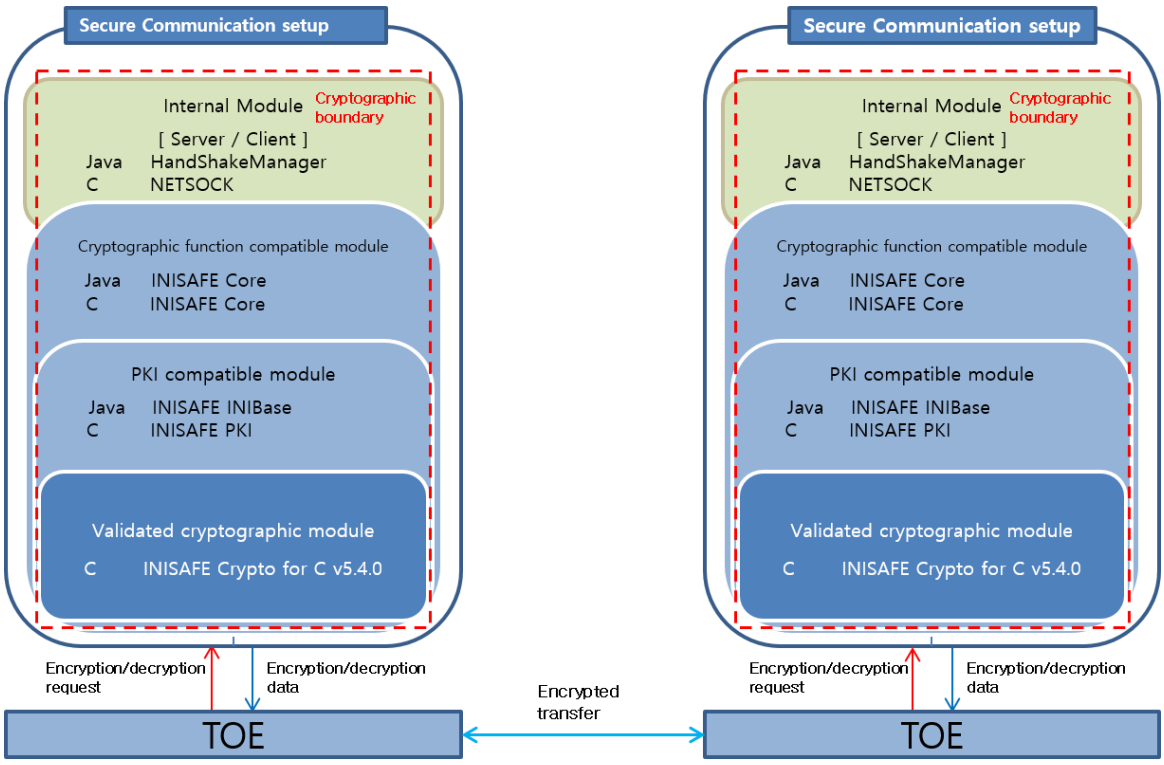
SFR to be satisfied: FMT_PWD.1(Extended)

6.6. Protection of the TSF (FPT)

6.6.1. Basic internal TSF data transfer protection

The encryption of communication (secure communication) used in data transfer between TOE components is as follows:
This is a network-based S/W cryptographic module that uses an encryption algorithm of the KCMVP-validated cryptographic module, and includes server and client functions in a single internal module. Therefore, it is possible to set up a server and a client as needed, depending on a situation. TCP/IP is used as a communication protocol by default, and all parameters transferred (key [public key, private key, secret key], key parameter, plaintext, ciphertext, etc.) are encrypted (satisfying FPT_ITT.1).

A cryptographic boundary for each component of the cryptographic module is shown below:



(Figure 7) Cryptographic boundary for each component

The key management for communication between components is performed by means of handshake encryption method using the KCMVP-validated cryptographic module, and the encryption of communication among TOE modules and mutual authentication are performed. All TOE components have and use the same private certificate (2,048-bit public key) for secure communication.

[Table 54] Management of communication encryption key and algorithm used

Category	Item	Description
Key Management Method	handshake	Asymmetric key encryption method that encrypts data after determining a key promise (algorithm, encoding rule, etc.) between the server and the client
Session Key	Generation algorithm	HASH-DRBG-SHA256
	Encryption algorithm	RSA
Transferred data	Encryption algorithm	SEED/128/CBC SHA-256

For the generation of a session key, Random is generated as specified in [Table 21] Random bit

generator.
Refer to 6.4.2 for detailed mechanisms of the communication procedure using the handshake encryption method.

SFR to be satisfied: FPT_ITT.1

6.6.2. Basic protection of stored TSF data

The TOE encrypts and stores cryptographic keys (DEK, Master Key, One-Day Key, etc.), critical security parameters (IV value and other encryption option values), TOE set value (security policy value, environment configuration value), administrator/service ID and password, DBMS account information and so forth (satisfying FPT_PST.1(Extended)).

[Table 55] Protection method in storing cryptographic keys and critical security parameters

Module	Encryption Target	Algorithm and Operation Method	Key Used	Storage Location	Application Method
SafeDB Policy Server	Master Key, iv	ARIA/256/CBC	Password-based derivation key	File	1st encryption
	Master Key	RSA 2048	Private certificate public key	File	2nd encryption
SafeDB Agent	One-Day Key, iv	ARIA/256/CBC	Password-based derivation key	Memory	Encryption
	User Data Encryption Key (DEK)	ARIA/256/CBC	One-Day Key	Memory	Encryption

[Table 56] Security policy and account information encryption list

Category	Cryptographic Key	Encryption Method	Algorithm	Encryption List	Data Storage Location
SafeDB Policy Server	Master Key	Symmetric key encryption	ARIA (CBC)	All security policy information	Policy DB

SafeDB Policy Server	-	Hash	SHA-256	Administrator and Service ID and password	Policy DB
SafeDB Policy Server	dbpwd	Symmetric key encryption	ARIA (CBC)	Policy DB account information cipher	File
SafeDB Agent	One-Day Key	Symmetric key encryption	ARIA (CBC)	All security policy information	Memory

[Table 57] Configuration file encryption and integrity check algorithm and key

Category	Type	Algorithm	List of Standards	Generation Method
SafeDB Policy Server	Configuration file encryption	ARIA(CBC)	KS X 1213-1(2019) KS X 1213-2(2019) TTAK.KO-12.0271-Part1/R1(2016) TTAK.KO-12.0271-Part3(2017)	Key generation using HASH-DRBG-SHA256
	Integrity check	HMAC-SHA-2	KS X ISO/IEC 9797-2(2018) TTAK.KO-12.0330-Part2(2018)	
SafeDB Agent	Configuration file encryption	ARIA(CBC)	KS X 1213-1(2019) KS X 1213-2(2019) TTAK.KO-12.0271-Part1/R1(2016) TTAK.KO-12.0271-Part3(2017)	Key generation using HASH-DRBG-SHA256

	Integrity check	HMAC-SHA-2	KS X ISO/IEC 9797-2(2018) TTAK.KO-12.0330-Part2(2018)	
SafeDB Manager	Integrity check	HMAC-SHA-2	KS X ISO/IEC 9797-2(2018) TTAK.KO-12.0330-Part2(2018)	Key generation using HASH-DRBG-SHA256
SafeDB SDK	Integrity check	HMAC-SHA-2	KS X ISO/IEC 9797-2(2018) TTAK.KO-12.0330-Part2(2018)	Key generation using HASH-DRBG-SHA256
SafeDB Plug-in	Integrity check	HMAC-SHA-2	KS X ISO/IEC 9797-2(2018) TTAK.KO-12.0330-Part2(2018)	Key generation using HASH-DRBG-SHA256

[Table 58] Storage for configuration file encryption Key and integrity check file

Category		Algorithm	Cryptographic Key	Application Method	Storage Location
SafeDB Policy Server	Configuration file encryption key	PBKDF2	configkey	Encryption	File
	Integrity check file	HMAC-SHA-2	-	HMAC	File
SafeDB Agent	Configuration file encryption key	PBKDF2	configkey	Encryption	File
	Integrity check file	HMAC-SHA-2	-	HMAC	File
SafeDB SDK	Integrity check file	HMAC-SHA-2	-	HMAC	File

SafeDB Plug-in	Integrity check file	HMAC-SHA-2	-	HMAC	File

For SafeDB Policy Server, SafeDB Agent, and SafeDB Manager, the cryptographic key for the configuration file is protected by using a pair of RSA keys generated upon the initial installation as KEK.

Additionally, configuration files and execution files that do not include TOE set values, such as IP address, port information, the number of unsuccessful authentication attempts and integrity check interval, are not encrypted. A file whose corruption can affect the operation of the TOE is subject to the integrity check (refer to 6.6.4 for more details).

SFR to be satisfied: FPT_PST.1(Extended)

6.6.3. Testing of external entities

The TOE performs a normal operation test on external entities (mail server, DBMS) upon the initial startup. SMTP access is tested by using mail server information set by the administrator in order to determine the normal operation of the mail server. An access test is carried out to determine the normal operation of DBMS by using basic commands provided by the DBMS or by using API.

SFR to be satisfied: FPT_TEE.1

6.6.4. TSF self-tests and integrity tests

The TOE runs a suite of self-tests during initial start-up and periodically (default: 60 minutes) during normal operation to demonstrate the correct operation as specified in [Table 59] Items subject to TOE self-tests.
The TOE verifies the integrity by comparing hash values of TSF executable code and stored configuration file. Integrity test is conducted during initial start-up and periodically (default: 60 minutes) during normal operation.
Integrity violation or self-test failure is notified to the administrator via email.
(satisfying FPT_TST.1)

[Table 59] Items subject to TOE self-tests

Category	Item	Content (Role)
SafeDB Policy Server	Cryptographic module	Self-test

	Process	Determine whether it was started normally at the time of start-up, and generate audit logs
SafeDB Agent	Cryptographic module	Self-test
	Process	Determine whether it was started normally at the time of start-up, and generate audit logs Confirm normal operation on a periodic basis and send the status to Policy Server
SafeDB SDK	Cryptographic module	Self-test
SafeDB Plug-In	Cryptographic module	Self-test
SafeDB Manager	Cryptographic module	Self-test

[Table 60] Items subject to TOE integrity test

Category	Item	Content (Role)
SafeDB Policy Server	Environment configuration file	TOE environment configuration file
	Stored TSF execution code	TOE process
SafeDB Manager	All files	Files composing the TOE
SafeDB Agent	Environment configuration file	TOE environment configuration file
	Stored TSF execution code	TOE process
SafeDB SDK	Environment configuration file	TOE environment configuration file
	Stored TSF execution code	TOE execution module
SafeDB Plug-In	Environment configuration file	TOE environment configuration file
	Stored TSF execution code	TOE execution module

TOE configuration file is protected against unauthorized modification as follows:

[Table 61] Protection of TOE configuration file against unauthorized modification

Category	Method for Modifying Configuration File	How to Find Modification
SafeDB Policy Server	The administrator executes MasterConsole function in SafeDB Policy Server, decrypts the configuration file, and modifies the content. After the modification, he/she executes	Newly generate HMAC-SHA-256 value of the configuration file upon initial start-up and on a periodic basis, and

	MasterConsole function again to encrypt the configuration file, and generates and stores integrity check value.	compare if it equals the stored value
SafeDB Agent	The administrator executes MasterConsole function in SafeDB Agent, decrypts the configuration file, and modifies the content. After the modification, he/she executes MasterConsole function again to encrypt the configuration file, and generates and stores integrity check value.	Newly generate HMAC-SHA-256 value of the configuration file upon initial start-up and on a periodic basis, and compare if it equals the stored value
SafeDB Manager	The administrator executes MasterConsole function in SafeDB Policy Server, decrypts the configuration file, and modifies the content. After the modification, he/she executes MasterConsole function again to encrypt the configuration file, and generates and stores integrity check value.	Newly generate HMAC-SHA-256 value of the configuration file upon initial start-up and on a periodic basis, and compare if it equals the stored value

Application notes: The configuration file is in plaintext upon the initial installation.

The following actions shall be taken if any abnormality occurs in self-test and integrity check items.

[Table 62] Actions in case of abnormality in TOE self-test items

TOE Module	Item	Check Time	Action
SafeDB Policy Server	Cryptographic module	Upon initial start-up	Send an email to the registered administrator and stop the start-up
		Periodically	Send an email to the registered administrator
SafeDB Agent	Cryptographic module	Upon initial start-up	Send an email to the registered administrator and stop the start-up
		Periodically	Send an email to the registered administrator
SafeDB SDK	Cryptographic module	Upon initial start-up	Transfer it to SafeDB Agent; send an email to the registered administrator; and stop the start-up
		Periodically	Transfer it to SafeDB Agent, and send an email to the registered administrator
SafeDB Plug-In	Cryptographic module	Upon initial start-up	Transfer it to SafeDB Agent; send an email to the registered administrator; and stop the

			start-up
		Periodically	Transfer it to SafeDB Agent, and send an email to the registered administrator
SafeDB Manager	Cryptographic module	Upon initial start-up	Send an email to the registered administrator and stop the start-up
		Periodically	Send an email to the registered administrator

[Table 63] Actions in case of abnormality in TOE integrity test items

TOE Module	Item	Check Time	Action
SafeDB Policy Server	Configuration file	Upon initial start-up	Send an email to the registered administrator, and stop the start-up
		Periodically	Send an email to the registered administrator
	Stored TSF executable code	Upon initial start-up	Send an email to the registered administrator, and stop the start-up
		Periodically	Send an email to the registered administrator
SafeDB Manager	Configuration file	Upon initial start-up	Send an email to the registered administrator, and stop the start-up
		Periodically	Send an email to the registered administrator
SafeDB Agent	Configuration file	Upon initial start-up	Send an email to the registered administrator, and stop the start-up
		Periodically	Send an email to the registered administrator
	Stored TSF executable code	Upon initial start-up	Send an email to the registered administrator, and stop the start-up
		Periodically	Send an email to the registered administrator
SafeDB SDK	Stored TSF executable code	Upon initial start-up	Transfer it to SafeDB Agent; send an email to the registered administrator; and stop the start-up
		Periodically	Transfer it to SafeDB Agent, and send an email to the registered administrator
SafeDB Plug-In	Stored TSF executable code	Upon initial start-up	Transfer it to SafeDB Agent; send an email to the registered administrator; and stop the start-up
		Periodically	Transfer it to SafeDB Agent, and send an email to the registered administrator

SFR to be satisfied: FPT_TST.1

6.7. TOE access (FTA)

6.7.1. TOE access

The TOE provides the capability to restrict the administrator's management access sessions, based on access IP that belongs to the same TSF administrator. The TOE provides the default value of two for the number of accessible IP.

It enforces the limitation on the maximum number of concurrent sessions of the administrator to one by default, and terminates the existing session in case of concurrent access.

If there is no interaction for the period of user inactivity (default value: 10 minutes), the TOE checks the period of inactivity, makes a request to WAS, and then terminates the session with the support of WAS.

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