INITECH
1st Financial & Security Service

Security Target (ST)

SafeDB

INITECH Co., Ltd.

Revision History

Ver	Date	Author	Revision
1.0	Mar. 22, 2018	Yongsik Gong	Initial release
1.1	May 18, 2018	Yongsik Gong	Incorporation of 1st TTA review result regarding EOR update
1.2	Jun. 14, 2018	Yongsik Gong	Incorporation of 2nd TTA review result
1.3	Jul. 6, 2018	Wonjoon Kim	Incorporation of the review findings
1.4	Jul. 16, 2018	Daehyeon Kim	Incorporation of the review findings
1.5	Nov. 21, 2018	Daehyeon Kim	Incorporation of the review findings
1.6	Feb. 14, 2019	Jiwoo Lee	Incorporation of the review findings

Table of Contents

List of Figures

List of Tables

1. ST Introduction

This chapter introduces the Security Target (ST) of SafeDB V4.0 of INITECH Co., Ltd.

1.1. ST reference

[Table 1] ST reference

Title	Security Target (ST)
Version	V1.6
Developer	Kim Daehyeon of INITECH CO., Ltd.
Publication Date	February 14, 2019
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_SB40_Security Target (ST)_V1.6
Product Classification	DB Encryption
Keywords	Database, Encryption

1.2. TOE reference

[Table 2] TOE reference

TOE Identification	SafeDB V4.0
TOE Version	V4.0 (Version detail: V4.0.3)
TOE Developer	INITECH Co., Ltd.
TOE Component	SafeDB Policy Server V4.0.3 SafeDB Manager V4.0.3 SafeDB Agent V4.0.3 SafeDB SDK for C V4.0.3 SafeDB SDK for Java V4.0.3 SafeDB Plug-In V4.0.3
Guidance Document	CCP.C_SB40_Preparative Procedure(PRE)_V1.4 CCP.C_SB40_Operational Guidance(OPE)_V1.4
Final Release	February 14, 2019

1.3. TOE overview

SafeDB V4.0 (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 was designed to be safe against hacking attacks such as sniffing with the network among the product modules encrypted. In addition, it is equipped with a mechanism to manage keys used for encryption in respect of the security.

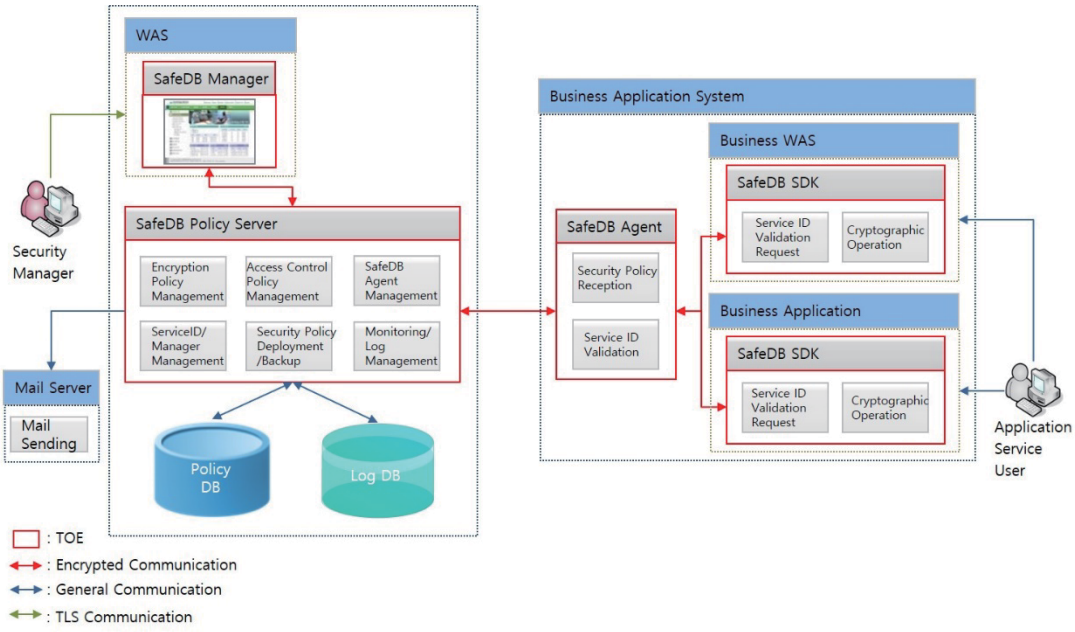
The TOE consists of SafeDB Policy Server that manages cryptographic operation policies and keys; SafeDB Manager that performs the web security management; SafeDB Agent installed and operated on the business server; SafeDB SDK (SafeDB SDK for C, SafeDB SDK for Java) that supports Java/C language; and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

If a 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.

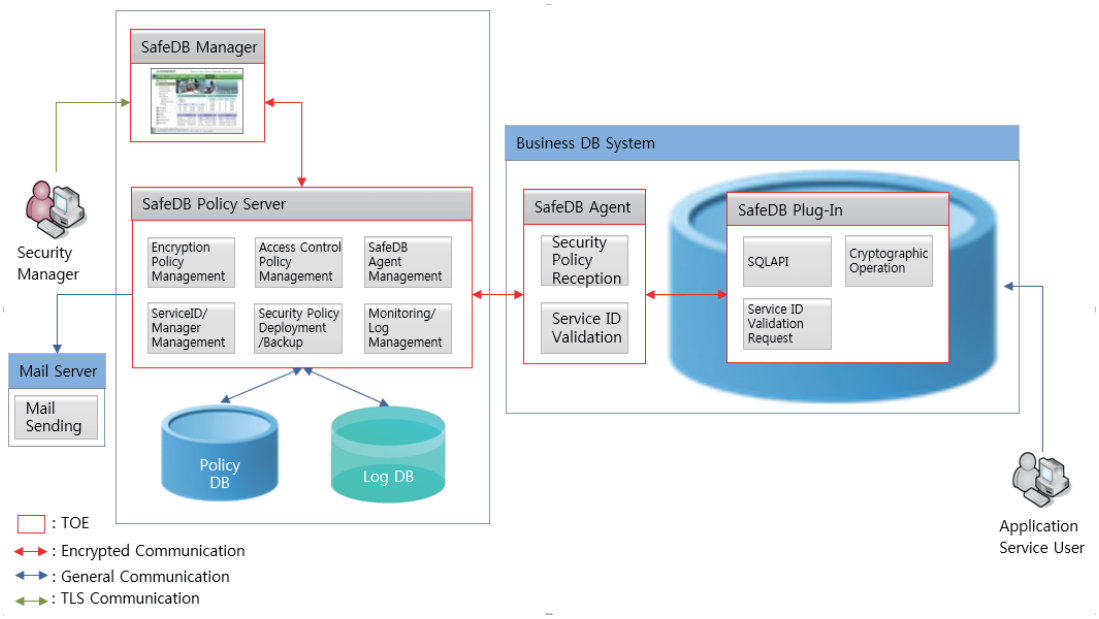
The TOE can be classified into two types: API type, Plug-In type.

API type encryption uses SafeDB SDK in the form of API on a business application and is irrelevant to 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) Block diagram of the TOE (API type)

The Plug-In method is an encryption method used by installing the SafeDB Plug-In module in the DBMS and is used with the same meaning as the SQL-API method. The SQL-API method is a method in which an application service user inputs an encryption / decryption function into a query (SQL statement).



(Figure 2) Block diagram of the TOE (Plug-In type)

1.3.1. TOE components and major security features

The TOE is a DB encryption solution that can protect important data by 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 offers, at the same time, both of the 1) SafeDB Plug-In type that can provide the DB security service only by installing it additionally in the DB without modifying the application nor undertaking a separate development process and 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, which allows for a DB security solution suitable for a circumstance of an individual customer.

The TOE consists of SafeDB Policy Server that manages encryption policies 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. Encryption policy management

		- Encryption policy and key management 2. SafeDB Agent management - Management of SafeDB Agent addition/deletion 3. Administrator and service ID management - Administrator identification and authentication - Management of administrator addition/deletion - Management of service ID addition/deletion 4. Security policy deployment - Automatic or manual deployment of security policies to SafeDB Agent 5. Monitoring/log management - SafeDB Agent status monitoring - Cryptographic operation status monitoring, etc. [Non-TSF] 1. Access control policy management 2. Security policy backup management, etc.
SafeDB Manager	Type	Software
	Function	[TSF] 1. Web-based security management - Registration/modification/deletion of administrator - Registration/modification/deletion of cryptographic operation policy - Registration/modification/deletion of SafeDB Agent - Registration/modification/deletion of Service ID - Audit log view - Real-time check on CPU/memory use of SafeDB Policy Server and Agent Server, etc.
SafeDB Agent	Type	Software
	Function	[TSF] 1. Security policy reception - Generation of One Day Key upon every start-up - Security policy reception and parsing - Storage of policy information encryption

		2. Service ID validation - Validation of Service ID requested from SafeDB SDK [Non-TSF] 1. Access control privilege check
SafeDB SDK	Type	Software
	Function	[TSF] 1. Service ID validation request and policy reception - One Day Key generation - Validation request per Service ID and policy reception - Storage of cryptographic key information encryption 2. Cryptographic operation - Cryptographic operation processing 3. Log generation - Audit log generation
SafeDB Plug-In	Type	Software
	Function	[TSF] 1. Data encryption/decryption in SQL API registered in DB - When using the query (SQL statement) Using the SafeDB encryption / decryption function registered in the DB - Internally, it is operated based on the logic same as SafeDB C SDK. Therefore, refer to the SafeDB SDK function above.

Security policies 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. ElasticSearch is used for the log DB.

[Table 4] Policy/log DB

Classification	Type	Description	Remarks
Policy DB	Apache Derby	File DB	V10.14.2.0
Log DB	ElasticSearch	File DB	V2.4.5

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

Type	Requirements for non-TOE operational environment					
	SafeDB Policy Server	**SafeDB Manager**	**SafeDB Agent**	**SafeDB Plug-In**	**SafeDB SDK for Java**	**SafeDB SDK for C**
OS	Windows Server 2008 R2 Enterprise SP1 64bit					
CPU	Intel Core i5-5200U 2.20GHz or higher					
Memory	8GB or higher					
HDD	40M or higher necessary for installing the TOE	70M or higher necessary for installing the TOE	10M or higher necessary for installing the TOE			
NIC	10/100/1000 X 1Port or higher					
Mandatory SW	ElasticSearch v2.4.5, Apache Derby v10.14.2.0 JRE 8.0(1.8.0_192)	JRE 8.0(1.8.0_192) Apache Tomcat v8.5.37,	JRE 8.0(1.8.0_192)	Oracle 12c	JRE 8.0(1.8.0_192)	

[Table 6] Encryption module used in the TOE

Classification	KCMVP Information	Remarks
INISAFE Crypto for Java v4.1	• Validation date: Dec. 16, 2014 • Validation No.: CM-97-2019.12	Used in Java module (SafeDB Policy Server, SafeDB Manager, SafeDB Agent SafeDB SDK for Java)

INISAFE Crypto for C V5.3.0	• Validation date: Jun. 15, 2018 • Validation No.: CM-137-2023.6	Used in C module (SafeDB Plug-In, SafeDB SDK for C)

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

Classification	Description
Mail Server	3rd Party Mail Server A link is established to send an alarm email to an administrator in case 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.
WAS (Web Application Server)	Apache Tomcat v8.5.37 supported Java-environment and web-based dynamic application execution server to operate SafeDB Manager for the identification, authentication and security management of the TOE
Web Client	Web browser is used to access/use SafeDB Manager on an administrator PC. Microsoft Internet Explorer 11 Chrome 65.0
DBMS	Policy DB, log DB Refer to [Table 4] Policy/log DB.

[Table 8] Requirements for administrator hardware

Classification	Item	Description
Admin PC	CPU	Intel Core2 2.13 GHz or higher
	RAM	2GB or higher
	HDD	500GB or higher
	NIC	10/100/1000 X 1Port or higher
	OS	Windows 7 Professional 64bit SP1
	Mandatory SW	Web Client in [Table 7] IT operational environment for implementing security features of the TOE

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 managing administrator and service ID and 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 an administrator; SafeDB Agent installed on each business server to receive the security policy and validate service ID; SafeDB SDK in charge of actual cryptographic operation and log recording; and SafeDB Plug-In installed on the DBMS in the form of plug-in(Cryptographic operation and logging).

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 composition of the TOE

Composition	Type	Identification Information			Within the TOE Scope
CD-ROM (1EA)	S/W	TOE installation program	SafeDB Policy Server	SafeDB_PolicyServer_4.0.3.zip	O
			SafeDB Manager	SafeDB_Manager_4.0.3.zip	O
			SafeDB Agent	SafeDB_Agent_4.0.3.zip	O
			SafeDB Plug-In	SafeDB_Plugin_4.0.3.zip	O
			SafeDB SDK for C	SafeDB_SDK_C_4.0.3.zip	O
			SafeDB SDK for Java	SafeDB_SDK_Java_4.0.3.zip	O
		3rd Party software	ElasticSearch v2.4.5		X
			Apache Derby v10.14.2.0		X
			Apache Tomcat 8.5.37		X
	Electronic document (PDF)	Guidance document	CCP.C_SB40_Preparative Procedure(PRE)_V1.4.pdf		O
			CCP.C_SB40_Operational Guidance(OPE)_V1.4.pdf		O

Certificate	Certificate	Software license certificate	X

The 3rd party software included of the product package CD are as follows:

- Apache Tomcat 8.5.37

This is a web-based dynamic application execution server in Java environment to operate SafeDB Manager.

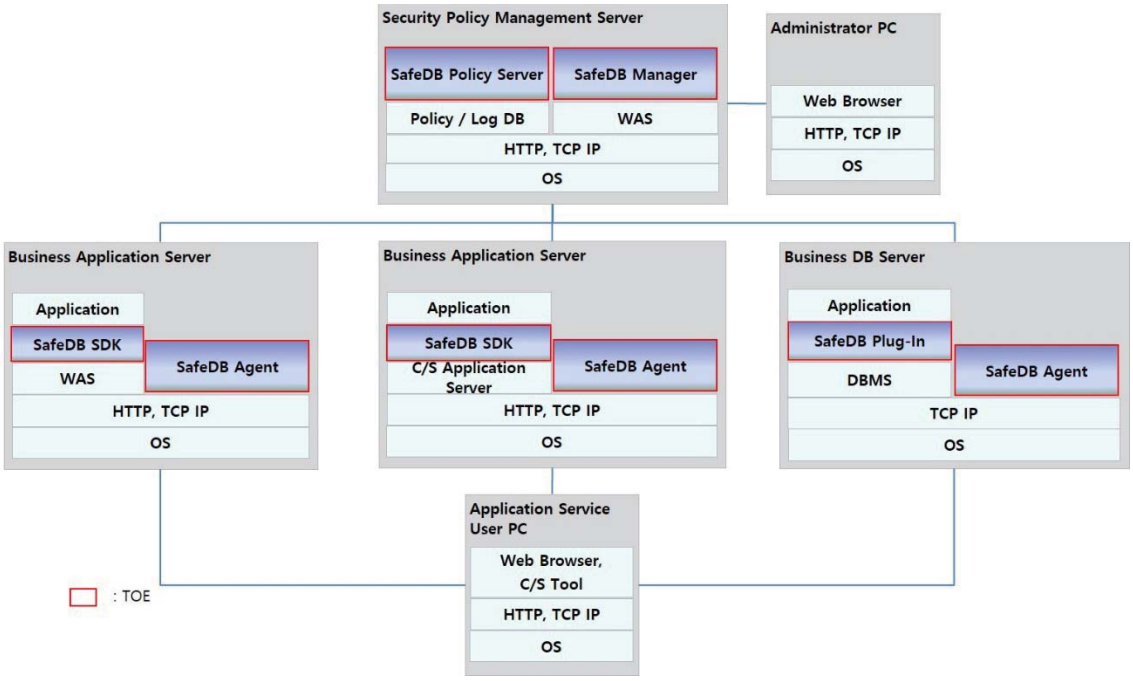
- Elasticsearch v2.4.5

This is a daemon for log storage and used when storing Plug-In Log and SDK Log in the "Log Name" in [Table 32] TOE log type.

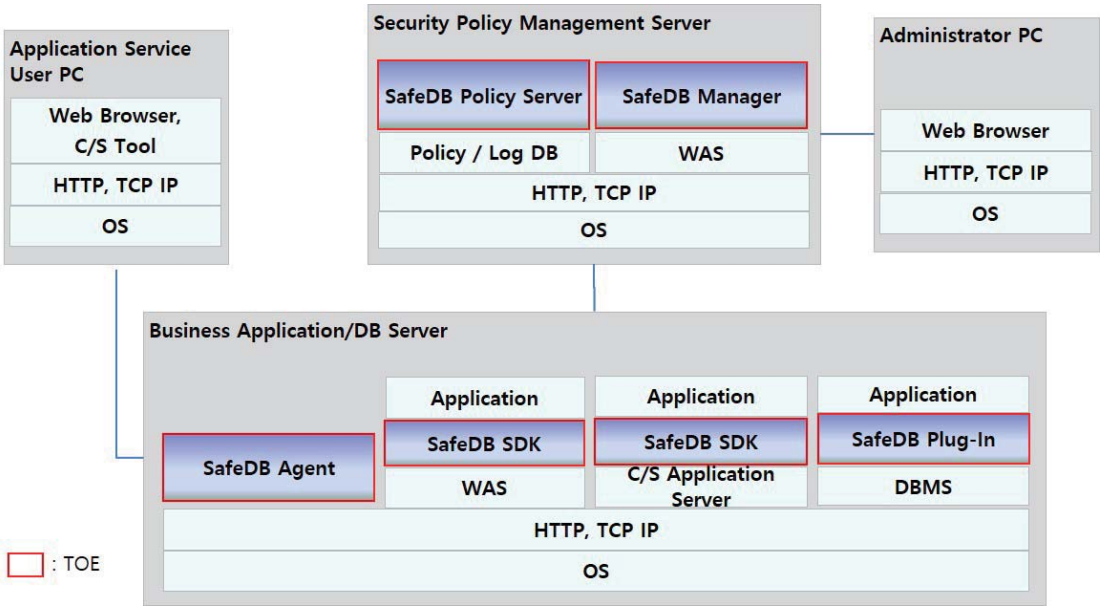
- Apache Derby v10.14.2.0

This is used to store the TSF data (encryption policy, etc.) generated when SafeDB Policy Server is operated.

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



(Figure 3) Physical scope of the TOE (1)



(Figure 4) Physical scope of the TOE (2)

It is a common practice to install SafeDB Policy Server and SafeDB Manager on the Security Policy Management Server and then for the remaining TOE modules, to install necessary modules on each business server as indicated in the Physical scope of the TOE (1). Depending on a situation, however, it is also possible to install the remaining TOE modules on one business server as shown in the Physical scope of the TOE (2).

As indicated above, the TOE includes SafeDB Policy Server, SafeDB Manager, SafeDB Agent, SafeDB Plug-In and SafeDB SDK, which are the software developed by INITECH Co., Ltd., as well as the preparative procedure and the operational guidance document for users. The CD also provides mandatory 3rd party software such as Apache Tomcat and ElasticSearch for the convenience of an operator.

Non-TOE operational environments such as hardware, operating system, web application server, database, java runtime environment, SSL environment, management console for an administrator to control SafeDB Policy Server and web browser using SafeDB Manager are excluded from the physical scope of the TOE.

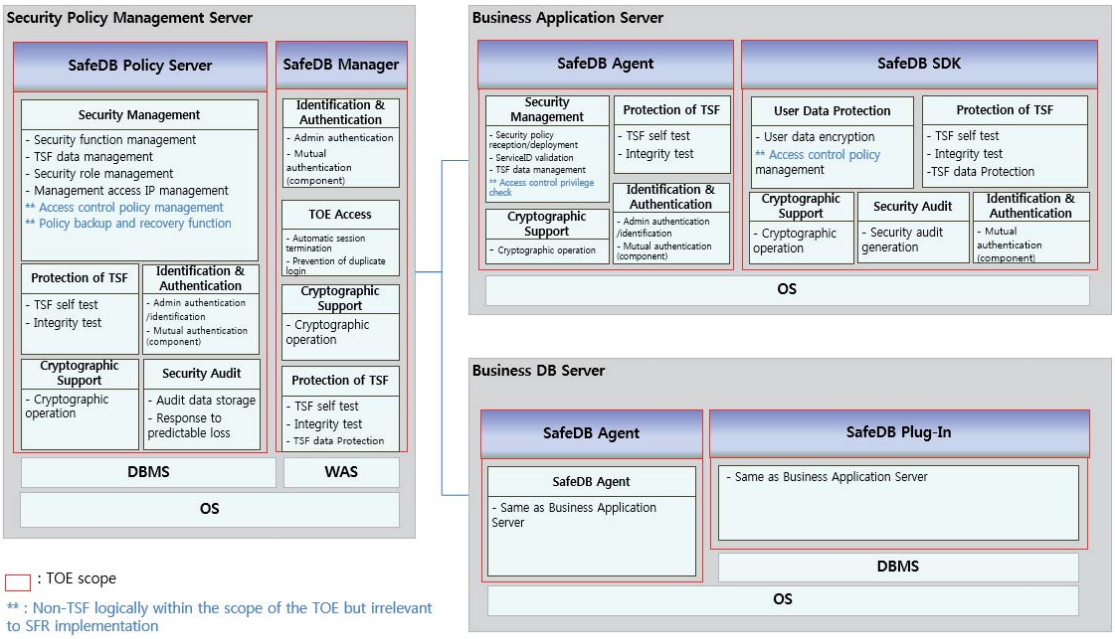
Management access of an administrator is provided via communication between a web browser on an administrator PC and the web server (Apache Tomcat), which is an operating environment of SafeDB Manager. TLS v1.2 encryption protocol is used to ensure a secure path.

1.4.2. TOE delivery method

The product package consists of CD 1EA and a certificate as shown in the [Table 9] Physical

composition 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 for setting and management of security functions provided by the TOE, TSF data and so forth.

- Security Management Function: It enables an 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. TSF data 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 Console Manager. Policy Manager can perform all web security management functions while General Manager can perform part of the web security management. Console Manager performs the security management through Master Console.

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

- Protection of the TSF

The TOE performs TSF self tests and conducts integrity test 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 DBMS and other external entities that interact with the TOE.

- Cryptographic Support

The TOE generates a cryptographic key for user data encryption and distributes the cryptographic key by using the KCMVP (Korea Cryptographic Module Validation Program)-validated cryptographic module. 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 supports the cryptographic operation function using the KCMVP validated cryptographic module such as 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 an 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 five minutes) so that the TOE is protected against adverse attempts of user authentication. 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 such as a reason for authentication failure during the authentication. 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 the operating environment of SafeDB Policy Server, and authorized administrator can review audit records. Furthermore, if the audit data reaches the threshold (fixed value of 90 percent), it is notified to the administrator via email. If an audit trail fills up (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 violations and takes actions to respond

to a detected 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

The TOE terminates the session by requesting the WAS after the manager inactivity period (fixed value of 10 minutes). In addition, it separately manages login information in order to prevent duplicated 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 test. 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: It performs the management of security functions. The TOE provides the function of receiving security policies assigned to SafeDB Agent and validating 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 a Console 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 test. 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).

- Cryptographic Support

The TOE 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 by using the validated cryptographic module.

4) SafeDB SDK

- Security Management

The security management function manages the security function and the TSF data.
- Security Function Management: It performs the management of security functions. The TOE provides the function of receiving security policies assigned to each service ID, based on which the cryptographic operation function is performed.
- TSF Data Management: The TOE manages the TSF data such as cryptographic keys and security policies.

- User Data Protection

The TOE performs the encryption or decryption of user data by using cryptographic operations. One Day Key is used in SafeDB Agent and SafeDB SDK, respectively, to encrypt and store a user data encryption key. Furthermore, the TOE ensures that any previous information content of a resource is made unavailable upon the allocation to and deallocation of the resource from user data.

- Protection of the TSF

The TOE performs TSF self tests and conducts integrity test. 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 on. Regarding the time used in audit records, timestamps are provided by the reliable operating system.

- Cryptographic Support

The TOE 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 by using the validated cryptographic module.

- 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 security function same 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 cryptographic keys, security policies, etc.

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

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

[SafeDB Plug-In]

- User Data Protection: It provides the function of user data encryption and decryption.

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 a subject)

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

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 validation 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.

SDK (Software Development Kit)

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

Master Console

A program that provides a function to execute preliminary work for driving the SafeDB Policy Server and SafeDB Agent.

1.7. ST organization

This document is structured as below:

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

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

Chapter 3 defines security problems and describes threats, organizational security policies and assumptions.

Chapter 4 Security Objectives describes the security objectives of the TOE and the operational environment and the rationale of the security objectives.

Chapter 5 describes the definition of extended components.

Chapter 6 Security Requirements describes the security functional requirements, the security assurance requirements, the security requirements and the rational of dependencies.

Chapter 7 provides the TOE summary specification.

2. Conformance Claim

This ST claims to conform to the followings:

2.1. CC conformance claim

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)
Conformance Claim	Part 2 Security Functional Requirements	Extended: FCS_RBG.1, FIA_IMA.1, FDP_UDE.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 PP for Database Encryption V1.0."

- PP Title and Version: National PP for Database Encryption V1.0
- Certificate No/Date: KECS-PP-0820-2017/2017-08-18
- Publication Date: 2017-08-18
- 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 EAL 1 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 PP for Database Encryption V1.0."

[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: OE augmented with conformance to PP selection SFR FPT_STM.1 requirement
  - OE.TRUSTED PATH: OE 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. The database shall not be generated, modified or deleted without a request from the TOE.

OE.TIME_STAMP	The TOE shall accurately record security-relevant events by using reliable time stamps provided by the TOE operational environment.
OE.TRUSTESD_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 TOE internal mutual authentication	1

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

Management: FIA_IMA.1
There are no management activities foreseen.

Audit: FIA_IMA.1
The following actions are recommended to record if FAU_GEN Security audit data generation family is included in the PP/ST:

a) Minimal: Success/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 User data encryption	1

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 function for [selection, choose one of: *setting ID and password when installing, setting password when installing, changing ID and password when the authorized administrator accesses for the first time, changing the password when the authorized administrator accesses for the first time*].

4.5. Protection of the TSF (FPT)

4.5.1. Protection of 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.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 sessions.

Component Leveling

In CC Part 2, the session locking and termination family consists of four components. In the National PP for Database Encryption V1.0, 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 session 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 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

Describes the functional and assurance requirements to be satisfied by the TOE that conforms to this Security Target.

5.1. Security functional requirements

All subjects, objects, operations, security attributes, and external entities used in the security requirements of this Security Target are defined as follows.

The subjects, objects, operations, and security attributes used in the ST security requirements refer to the SFRs, and external entities refer to Chapter 1.

The security requirements of this Security Target comprise the functional components of CC Part 3 (CC V3.1). The security functional components are summarized as follows.

[Table 11] Summary of security functional components

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

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

5.1.1. 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.

[Table 12] Actions against security violations

Security Violations Events	Response Actions
Accumulation of authentication failure events specified in FIA_UAU.2	· Limitation on login attempts for a specified period of time (fixed value of 5 minutes) for all authorized administrators · Sending a warning email to the administrator
The integrity violation event specified in FPT_TST.1 and the self-test failure event of the validated cryptographic module	· Sending a warning email to the administrator

Audit trail specified in FAU_STG.3 exceeded the disk capacity Case	· Sending 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.

c) [*Refer to the "auditable events" of 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 [*Refer to the contents of "Additional audit record contents" in [Table 13] Auditable events*, *None*]

[Table 13] Auditable events

Security functional component	Auditable events	Additional audit record contents
FAU_ARP.1	Actions taken due to potential security violations	
FAU_SAA.1	Enabling and disabling of any of the analysis mechanisms, Automated responses performed by the tool	
FAU_STG.3	Actions taken due to exceeding of a threshold	
FAU_STG.4	Actions taken due to the audit storage failure	
FCS_CKM.1(1)	Success and failure of the activity	
FCS_CKM.2	Success and failure of the activity (only applying to distribution of key related to user data encryption/decryption)	
FCS_CKM.4	Success and failure of the activity (only applying to destruction of key related to user data encryption/decryption)	
FCS_COP.1(1)	Success and failure of the activity	
FDP_UDE.1	Success and failure of user data	

	encryption/decryption	
FIA_AFL.1	The reaching of the threshold for the unsuccessful authentication attempts and the actions taken, and the subsequent, if appropriate, restoration to the normal state	
FIA_IMA.1	Success and failure of mutual authentication Modify of authentication protocol	
FIA_UAU.2	Any use of authentication mechanisms	
FIA_UAU.4	Attempts to reuse authentication data	
FIA_UID.2	All use of the user identification mechanism, including the user identity provided	
FMT_MOF.1	All modifications in the behavior of the functions in the TSF	
FMT_MTD.1	All modifications to the values of TSF data	Modified values of TSF data
FMT_PWD.1	All changes of the password	
FMT_SMF.1	Use of the management functions	
FMT_SMR.1	Modifications to the user group of rules divided	
FPT_TST.1	Execution of the TSF self test 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 external physical test and test results	

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,

[audit store capacity exceeded during audited events in FAU_STG.3] known to indicate a potential security violation

b) [Other audit event rules including potential violations]

[

- FIA_UAU.2 : five duplicate authentication failures occurred during an administrator's authentication failure audit
- FIA_STG.3 : limit of the audit trail exceeds 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 [*methods of selection and/or ordering*] of audit data based on [*criteria with logical relations*].

[

- criteria with logical relations : Combination with And operation when entering the selected reference value in *[Table 14] Audit data type and selection criteria*.
- methods of selection and/or ordering : Select audit data according to *[Table 14] Audit data type and selection criteria* and order audit data in descending order based on time of occurrence.

]

[Table 14] Audit data type and selection criteria

Audit data type	Selection Criteria	Permissibility
Plug-In Log (Plug-In Service Log)	Incident date (Start ~ End)	Search, Sort (in the order of time of occurrence)
	SID name	
	Service ID	

	Service ID's connect IP	
	Service ID's Mac	
	Owner name	
	Table name	
	Column name	
	Log message	
SDK Log (SDK Service Log)	Incident Date (Start ~ End)	Search, Sort (in the order of time of occurrence)
	SDK name	
	HOST name	
	Service ID's connect IP	
	Service ID	
	Connected SafeDB Agent's IP : Port	
	Function (Login, Encrypt, Decrypt, Logout)	
	Log level (ERROR, DEBUG, INFO)	
	Table name	
	Column name	
	Policy name	
	Log message	
Manager Log	Incident date (Start ~ End)	Search, Sort (in the order of time of occurrence)
	Manager ID	
	Log type (Entire, Common resource management, Policy management)	
	Manager name	
	Log content	
	Manager ID	
	Common DB name	
	Policy name	
	Table name	
	Column name	
Mail Log (Mail Log Sent to Administrator)	Incident date (Start ~ End)	Search, Sort (in the order of time of occurrence)
	Log content	
Console Log (Log generated by MasterConsole)	Period	Search, Sort (in the order of time of occurrence)
	Type	
	ID	
	Name	
	IP	

	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 should take [*Notify the authorized administrator*, [*None*]] if the audit trail exceeds the [*percentage of free space to the total capacity of the audit record storage space*].

Application caution: Notification to the authorized administrator is carried out by sending mail. The TSF carries out the retransmission of the audit data to prevent loss of the audit data if the audit trail fails to transmit to Log DB from the SafeDB SDK, SafeDB Plug-In which is the TOE module.

FAU_STG.4 Prevention of audit data loss

Hierarchical to FAU_STG.3 Action in case of possible audit data loss

Dependencies FAU_STG.1 Protected audit trail storage

FAU_STG.4.1 The TSF should *ignore the audit event* and execute [*Send mail to an approved 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 Number Generation

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

[Table 15] User data encryption algorithm and key sizes

Standard list	Cryptographic operation	Cryptographic algorithm	complexity of cryptographic	Cryptographic key sizes	Purpose
KS X 1213	Symmetric key encryption	ARIA (CBC)	256	256	Encrypt user data (symmetric key ciphering)
TTAS.KO-12.0004	Symmetric key encryption	SEED (CBC)	128	128	
ISO/IEC 10118-3	Secure Hash	SHA-2	256	None	Encrypt user data (HASH)
			512	None	
ISO/IEC 9797-2	Message authentication code	HMAC-SHA-2	256	512	Encrypt user data (HMAC)
			512	1024	
TTAK.KO-12.0191	Random bit generator	HASH-DRBG-SHA256	256		DEK for creation

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

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

Hierarchical to: No other components

Dependencies: [FCS_CKM.2 Cryptographic key distribution, or
FCS_COP.1 Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction
FCS_RBG.1 Random Number Generation

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

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

Standard list	Cryptograp hic	Crypt ograp	Cryptogra phic key	TOE Module	Describe the key type and	Cryptograph ic key

	operations	hic algorithm	sizes		purpose	Creation
KS X 1213	Symmetric key encryption	ARIA (CBC)	256	SafeDB Policy Server	Used to store security policy encryption with Master Key	Using the Random bit generator
ISO/IEC 10118-3	Secure Hash	SHA-2 (256)	None	SafeDB Policy Server	Administrator and Service ID Password hash	-
KS X 1213	Symmetric key encryption	ARIA (CBC)	256	SafeDB Agent	Used to store security policy encryption with One Day Key	Using the Random bit generator
KS X 1213	Symmetric key encryption	ARIA (CBC)	256	SafeDB SDK	One Day Key for storing security policies or cryptographic key encryption	Using the Random bit generator
TTAK.KO-12.0274	Password Based Key Derivation Functions	HMAC-SHA-2	256	SafeDB Policy Server	For encryption when saving cryptographic keys, such as master keys	Entered by the administrator
ISO/IEC 9797-2	Message authentication code	HMAC-SHA-2	256	SafeDB SDK	For examining log forging/corruption with Log Key	Using the Random bit generator
ISO/IEC 18033-2	Public Key encryption	RSAES	2048	SafeDB Policy Server	For encryption when storing Master Key / Log Key	Using the RSA key generation algorithm
ISO/IEC 18033-2	Public Key encryption	RSAES	2048	TOE Inter-module communication	Used Session Key encryption	Using the RSA key generation algorithm
TTAS.KO-	Symmetric	SEED	128	TOE	Used Session	Using the

12.0004	key encryption	(CBC)		Inter-module commu nication	Key for interval encrypt between TOE	Random bit generator
KS X 1213 ISO/IEC 9797-2	Symmetric key encryption/ Message authenticati on code	ARIA (CBC) HMAC -SHA-2	256	SafeDB Policy Server etc.	settings file encryption, integrity	Using the Random bit generator

Application caution : Master Key : Key used by SafeDB Policy Server to encrypt security policy, cryptographic key, One Day Key: SafeDB Agent, SafeDB SDK, SafeDB Plug_In to encrypt security policy and cryptographic key.

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 destruct cryptographic keys in accordance with a specified cryptographic key destruction method ["*Distribution method" of [Table 17] Cryptographic key distribution method*] that meets the following: ["S*tandard list" of [Table 17] Cryptographic key distribution method*].

[Table 17] Cryptographic key distribution method

Standard list	Distribution target	Distribution method
KS X ISO/IEC 11770-3:2008	DEK from FCS_CKM.1(1)	ommunication interval encryption using handshake encryption using validated cryptographic module
KS X ISO/IEC 11770-3:2008	Communication interval between TOE modules Encryption Session Key from FCS_CKM.1(2)	Handshake Encryption Using 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 18] Destruction according to cryptographic key storage*] that meets the following: [*"Standard list" of [Table 18] Destruction according to cryptographic key storage*].

[Table 18] Destruction according to cryptographic key storage

Standard list	Cryptographic key storage location	destruction method	Detailed method of destruction	Destruction object	Destruction point
None	DB	deletion	Delete SQL to delete from DB	Data encryption key Security policy information	When the administrator deletes the security policy
None	Memory	Memory zeroization	Overwrite everything with "0".	Master Key / One Day Key Log Key	When calling process shut-down or log-out function
		Initialize variables	Main variable null initialization processing	Security policy list	
None	Memory	Memory release	After all keys are overwritten to "0", memory release	Session Key	At the end of communication
None	Memory	Memory zeroization or initialization	Overwrite everything with "0" or null initialization processing	KEK	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 ["*list of cryptographic operations" of [Table 15] User data encryption algorithm and key*] in accordance with a specified cryptographic algorithm [*"Cryptographic algorithm" of [Table 15] User data encryption algorithm and key*] and cryptographic key sizes [*["Cryptographic key sizes" of [Table 15] User data encryption algorithm and key*] that meet the following: [*"Standard list" of [Table 15] User data encryption algorithm and key*].

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 [*"cryptographic operations" of [Table 16] TSF data cryptographic key algorithm and key*] in accordance with a specified cryptographic algorithm [*"cryptographic algorithm" of [Table 16] TSF data cryptographic key algorithm and key*] and cryptographic key sizes [*"cryptographic key sizes" of [Table 16] TSF data cryptographic key algorithm and key*] that meet the following: [*"Standard list" of [Table 16] TSF data cryptographic key algorithm and key*].

FCS_RBG.1 Random bit generation (Extended)

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RBG.1.1 The TSF shall generate random bits required to generate an cryptographic key using the specified random bit generator that meets the following [*[Table 19] Random bit generator "Standard list"*].

[Table 19] Random bit generator

Standard list	Random bit generator	Base function
TTA.KO-12.0191	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 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 method 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 [*Administrator authentication attempts*].

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

[

- Disable identification and authentication (fixed value of 5 minutes)
- Send alert mail to administrator

]

Application notes: If the administrator's authentication attempt occurs more than 5 times, the identification and authentication functions are disabled, so normal login can be

performed after 5 minutes. However, the policy administrator can unlock other accounts who have been account locked out.

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" which is an authentication protocol using the validated cryptographic module*] in accordance with [*[Table 20] Mutual verification among the TOE components "Mutual verification interval among the TOE components"*] between [*None*].

[Table 20] TOE intenral mutual authentication

Mutual verification interval among the TOE components		Validated cryptographic module
SafeDB Policy Server	SafeDB Manager	Refer to *[Table 6] Encryption 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 acceptance criteria].

[

a) Permitted characters

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

b) Minimum/Maximum length

- 9 ~ 16 digits

c) Combination rules

- combination of 3 or more among English alphabets/numbers/special characters
- Excluded if a password has three or more successive letters (numbers)
- Excluded if a password was used the last five times
- 3 special characters on the keyboard (',",;) are excluded

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

- 0 ~ 999 days (the interval is defined by the authorized administrator)

]

FIA_UAU.2 User authentication prior to all actions

Hierarchical to FIA_UAU.1 Timing of authentication

Dependencies FIA_UID.1 Timing of identification

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

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 [*identified authentication mechanisms*].

[

- TOE authenticates administrators with password-based password authentication.
- TOE stores and checks the authentication time and number of failures in the DB at each administrator's authentication.
- TOE checks for duplicate authentication requests by storing the credentials in memory each time the administrator authenticates.
- TOE prevents reuse of authentication data by storing and examining the only session ID in memory for password-based encryption.

]

FIA_UAU.7 Protected authentication feedback

Hierarchical to No other components.

Dependencies FIA_UAU.1 authentication

FIA_UAU.7.1 The TSF shall provide only [*'*' characters, no input characters displayed*] to the user while the authentication is in progress

FIA_UID.2 User identification prior to all actions

Hierarchical to FIA_UID.1 Timing of authentication

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 behavior

Hierarchical to No other components.

Dependencies FMT_SMF.1 Specification of Management Functions
FMT_SMR.1 Security roles

FMT_MOF.1.1 The TSF shall restrict the ability to ***conduct management actions of*** the functions [*[Table 21] List of security functions behavior of administrator*] to [*the authorized roles*].

[Table 21] List of security functions behavior of administrator

Administrator Type	Classification	Security Function	Ability			
			Determine the behavior	disable	enable	Modify the behavior
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	Monitoring	SafeDB Policy Server CPU/memory use	O	-	-	O
		SafeDB Agent CPU/memory use	O	-	-	O
	Additional management	Cryptographic operation status	O	-	-	-
		Audit data review	O	-	-	-
Console Manager	Initialization	TOE initialization setting	O	-	O	-
	Integrity validation	Configuration integrity validation file generation	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 23] *[Table 22] List of TSF data and management ability*] to [*authorized administrator*].

[Table 22] List of TSF data and management ability

Administrator Type	TSF data	Ability				
		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				
Console Manager	Security policy					O
	Cryptographic key (Master Key, Log Key, account information encryption key, etc.)				O	
	Audit data					O
	Set value		O			

Application caution: The console administrator must perform security policies, initialize audit data, and generate encryption keys such as Master Key for startup and operation of SafeDB Policy Server.

FMT_PWD.1 Management of ID and password(Extended)

Hierarchical to No other components.

Dependencies FMT_SMF.1 Specification of Management Functions

FMT_PWD.1.1 The TSF shall restrict the ability to manage the password of [*Manager 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 [*Manager management function, Service ID management function*] to [*None*].

1. [*None*]
2. [*None*]

FMT_PWD.1.3 The TSF shall provide the ability for the authorized administrator of the SafeDB Manager to change the password upon initial connection and to set the master administrator authorized administrator password during the installation process.

FMT_SMF.1 Specification of Management Functions

Hierarchical to No other components.

Dependencies No dependencies.

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: [*list of management functions to be provided by the TSF*].

[

- TSF Function Management : Management Functions as specified in FMT_MOF.1
- TSF Data Management : Management Functions as specified in FMT_MTD.1
- ID and password management : Management function as 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 23] Classification and roles of administrators*].

[Table 23] Classification and roles of administrators

Type	Level	Roles
Security Manager	Policy Manager	Perform Web security management functions (query, create, change, delete security policy)
	General Manager	Perform Web security management functions (query security policy)
Console Manager	-	Perform console security management functions

FMT_SMR.1.2 TSF 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 [TSF data] stored in containers controlled by the TSF from

the unauthorized *disclosure, modification*.

[*TSF data* :

- Encryption Policy : DB name, owner name, table name, column name, user Data Encryption Key (DEK), encryption options
- Agent information, common resource information
- Administrator account information
- Policy DB access account information
- Cryptographic key (Master Key, One Day Key, etc.)
- TOE set value (Number of authentication failures, integrity check interval, IP, port information, etc.)

]

FPT_TEE.1 Testing of external entities

Hierarchical to No other components.

Dependencies No dependencies.

FPT_TEE.1.1 The TSF shall run a series of tests at initial startup to check that the ["List" of *[Table 24] External entity attribute satisfies normal operating conditions*].

[Table 24] External entity attribute

List	Use
Mail Server	Send notification mail to authorized administrators
DBMS	Save security policy, save audit data

FPT_TEE.1.2 If the test fails, the TSF shall perform [*output error and stop running when DBMS test fails, error message output when mail server test fails*].

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 25] Items subject to TOE self test*].

[Table 25] Items subject to TOE self test

Classification	Item	Content (role)

SafeDB Policy Server	cryptographic module	self test
	process	Determine if the startup started normally and generate an audit log
SafeDB Agent	cryptographic module	self test
	process	Determine if the startup started normally and generate an audit log Periodically check agent behavior and send status to Policy Server
SafeDB SDK	cryptographic module	self test
SafeDB Plug-In	cryptographic module	self test
SafeDB Manager	cryptographic module	self test

Application notes: The period of self-testing periodically during normal operation is 60 minutes.

[Table 26] Items subject to TOE integrity test

Classification	Item	Content (role)
SafeDB Policy Server	Configuration file	TOE Configuration file
	Stored TSF executable code	Policy server daemon process
SafeDB Manager	Configuration file	TOE Configuration file
SafeDB Agent	Configuration file	TOE Configuration file
	Stored TSF executable code	Agent daemon process
SafeDB SDK	Configuration file	TOE Configuration file
	Stored TSF executable code	API behavior process
SafeDB Plug-In	Configuration file	TOE Configuration file
	Stored TSF executable code	Plug-In behavior process

Application notes: The integrity check cycle performed periodically during normal operation is 60 minutes.

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

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

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_SMF1.1]

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

b) limit the maximum number of concurrent sessions to {unlimited} 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 administrator's interactive session after a [*time interval 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 default value of the number of connectable IPs provided by the TOE is set to two.

5.2. Security assurance requirements

Assurance requirements of this Security Target are comprised of assurance components in CC part 3, and the evaluation assurance level is EAL1+(ATE_FUN.1). The following table summarizes assurance components.

[Table 27] Summary of Security Assurance Components

Security assurance class	Security assurance component	
Security Target evaluation	ASE_INT.1	ST introduction
	ASE_CCL.1	Conformance claims
	ASE_OBJ.1	Security objectives for the operational environment
	ASE_ECD.1	Extended components definition
	ASE_REQ.1	Stated security requirements
	ASE_TSS.1	TOE summary specification
Development	ADV_FSP.1	Basic functional specification
Guidance documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative procedures
Life-cycle support	ALC_CMC.1	Labelling of the TOE
	ALC_CMS.1	TOE CM coverage
Tests	ATE_FUN.1	Functional testing
	ATE_IND.1	Independent testing - conformance
Vulnerability assessment	AVA_VAN.1	Vulnerability survey

5.2.1. Security Target evaluation

ASE_INT.1 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 claims

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

Developer action elements

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

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

Content and presentation elements

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

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

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

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

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

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

ASE_CCL.1.7C The conformance claim rationale shall demonstrate that the TOE type is consistent with the TOE type in the PPs for which conformance is being claimed.

ASE_CCL.1.8C The conformance claim rationale shall demonstrate that the statement of the security problem definition is consistent with the statement of the security problem definition in the PPs for which conformance is being claimed.

ASE_CCL.1.9C The conformance claim rationale shall demonstrate that the statement of security objectives is consistent with the statement of security objectives in the PPs for which conformance is being claimed.

ASE_CCL.1.10C The conformance claim rationale shall demonstrate that the statement of security requirements is consistent with the statement of security requirements in the PPs for which conformance is being claimed.

Evaluator action elements

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

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 developer shall provide a statement of security requirements.

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

Content and presentation elements

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

ASE_REQ.1.2C All subjects, objects, operations, security attributes, external entities and other

terms that are used in the SFRs and the SARs shall be defined.

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

ASE_REQ.1.4C All operations shall be performed correctly.

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

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

Evaluator action elements

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

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_PRE1.1C 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_PRE1.2C 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 TOE Lavelling 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 meet 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 following: the TOE itself; and the evaluation evidence required by the SARs.

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

Evaluator action

elements

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

5.2.5. Tests

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 presentation of evidence.

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

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

5.3. Security requirements rationale

5.3.1. Dependency of the SFRs of the TOE

The security functional requirements used in this Security Target satisfies the dependency as shown in the following table, and no component does not satisfy the dependency relationship.

The following table shows dependency of security functional requirements.

[Table 28] Dependencies of TOE SFRs

No.	Security functional requirements	Dependency	Reference No.
1	FAU_ARP.1	FAU_SAA.1	3
2	FAU_GEN.1	FPT_STM.1	Rationale(1)
3	FAU_SAA.1	FAU_GEN.1	2
4	FAU_SAR.1	FAU_GEN.1	2
5	FAU_SAR.3	FAU_SAR.1	4
6	FAU_STG.3	FAU_STG.1	Rationale(2)
7	FAU_STG.4	FAU_STG.1	Rationale(2)
8	FCS_CKM.1(1)	FCS_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 a dependency on FPT_STM.1, but the security objectives OE of the operational environment of this ST. Since reliable time stamps provided by time stamps are used, they satisfy dependencies.

Rationale(2) : Although FAU_STG.3 and FAU_STG.4 have dependency on FAU_STG.1, the security objectives OE of the operational environment of this ST. Since it is protected from unauthorized deletion and change by a secure DBMS, it satisfies 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 of SARs of the TOE

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

The augmented SAR ATE_FUN.1 has dependency on ATE_COV.1. The rationale for the relevant dependencies is in accordance with the national database encryption protection profile V1.0. but, ATE_FUN.1 is augmented to

require developer testing in order to check if the developer correctly performed and documented the tests in the test documentation, ATE_COV.1 is not included in this PP 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 29] 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 prior to all actions
	FIA_UAU.4	Single-use authentication mechanisms
	FIA_UAU.7	Protected authentication feedback
	FIA_UID.2	User identification prior to all actions
Security Management (FMT)	FMT_MOF.1	Management of security functions behavior
	FMT_MTD.1	Management of TSF data

	FMT_PWD.1(Extended)	Management of ID and password
	FMT_SMF.1	Specification of management functions
	FMT_SMR.1	Security roles
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.

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 30] Actions against security violations in case potential security violation is detected, based on audit records generated (refer to 6.1.1).

[Table 30] 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 5 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) In case the threshold pre-defined by the administrator is reached (90 percent), the administrator is notified via email.

b) In case 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 means what percentage of the total capacity of HDD using log DB is in use. For example, if the threshold of 100 GB HDD is set as 90 percent, it means that 90 GB is in use and 10 GB is available for further use.

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 31] Audit data type and selection criteria, the search result that has a combination with AND operation can be displayed.

[Table 31] Audit data type 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 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 : Port of SafeDB Agent accessed	
	Function (Login, Encrypt, Decrypt, Logout)	
	Log level (ERROR, DEBUG, INFO)	
	Table name	
	Column name	
	Policy name	
	Log message	
Manager Log (Admin Log)	Event date and time (start date – end date)	Search, Sort (in the order of time of occurrence)
	Administer ID	
	Log type (all, common resource management, policy management)	
	Administrator name	

	Log content	
	Administrator ID	
	Common DB name	
	Policy name	
	Table name	
	Column name	
Mail Log (Log of sending emails to admin)	Event date and time (start date – end date)	Search, Sort (in the order of time of occurrence)
	Log content	
Console Log (Log generated in Master Console)	Period	Search, Sort (in the order of time of occurrence)
	Type	
	ID	
	Name	
	IP	
	Message	

[Table 32] TOE log type

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 function.

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 33] User data encryption key generation method

List of Standards	Encryption Method	Algorithm	Cryptographic Security	Use
TTAK.KO-12.0191	Random bit generator	HASH-DRBG-SHA256	256	For DEK 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 34] TSF data cryptographic key generation method below (satisfying FCS_CKM.1(2)).

[Table 34] TSF data cryptographic key generation method

List of Standards	Encryption Method	Algorithm	Cryptographic Security	Use
TTAK.KO-12.0191	Random bit generator	HASH-DRBG-SHA256	256	Generation of Master Key, One Day Key, Log Key and session key
TTAK.KO-12.0274	Password Based Key Derivation Functions	HMAC-SHA-2	256	Generation of cryptographic key encryption key such as Master Key
ISO/IEC 18033-2	Public key encryption	RSAES	2048	Generation of cryptographic key for Master Key / Log Key/session key encryption

The TOE uses HASH-DRBG-SHA256 random bit generator in generating cryptographic 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 security 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 35] 11.1 RSA key generation algorithm

Classification	Value	Content
Input	Size of modulus n |n|	Selection of a size of modulus subject to KCMVP validation | n | = 2048, 3072 bit
Output	Public key P_{ub} = (n, e) Private key P_{riv} = (n, d)	Refer to [Table 36] RSA key generation method

[Table 36] 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 19] 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 37] 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 a 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 a validated cryptographic module

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

[Table 38] 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 a 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 a 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 a 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 a 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).

One Day Key and the security policy information (including DEK) stored in the memory are destroyed when the administrator calls SafeDB Agent Process Shut-down command (satisfying FCS_CKM.4).

SafeDB SDK, SafeDB Plug-In destroys One Day Key and 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 on 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 39] Cryptographic key destruction method per storage

List of Standards	Cryptographic Key Storage	Destruction Method	Detailed Destruction Method	Destruction Target	Timing of Destruction
N/A	DB	Deletion	Execute Delete SQL to delete in the DB	Data encryption key	When the administrator deletes the security policy
				Security policy information	
N/A	Memory	Memory zeroing	Overwrite all the keys with "0"	Master Key / One Day Key Log Key	When calling process Shut-down or Log-out function
		Parameter initialization	Parameter initialization: initialization by nullifying major parameters	List of security policies	
N/A	Memory	Memory release	Overwrite all the keys with "0" and release the memory	Session Key	When terminating the communication
N/A	Memory	Memory zeroing or initialization	Overwrite all the keys with "0" or initialize by nullifying	KEK	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 lengths used in cryptographic operation of user data are as follows:

[Table 40] User data encryption algorithm and key sizes

List of Standards	Encryption Method	Algorithm	Cryptographic Security	Cryptographic Key Sizes	Use
KS X 1213	Symmetric key encryption	ARIA (CBC)	256	256	User data encryption (symmetric key cryptographic operation)
TTAS.KO-12.0004	Symmetric key encryption	SEED (CBC)	128	128	
ISO/IEC 10118-3	Secure Hash	SHA-2	256	N/A	User data encryption (HASH)
			512	N/A	
ISO/IEC 9797-2	Message authentication code	HMAC-SHA-2	256	512	User data encryption (HMAC)
			512	1024	
TTAK.KO-12.0191	Random bit generator	HASH-DRBG-SHA256	256	N/A	For DEK generation

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

[Table 41] TSF data encryption algorithm and key sizes

List of Standards	Encryption Method	Algorithm	Cryptographic Key sizes	TOE Module	Key Type and Use	Cryptographic Key Generation Method
KS X 1213	Symmetric key encryption	ARIA (CBC)	256	SafeDB Policy Server	Used when storing security policy encryption with Master Key	Using random bit generator
ISO/IEC 10118-3	Secure Hash	SHA-2 (256)	N/A	SafeDB Policy Server	Administrator and service ID password hash	-
KS X 1213	Symmetric key encryption	ARIA (CBC)	256	SafeDB Agent	Used when storing security policy	Using random bit generator

					encryption with One Day Key	
KS X 1213	Symmetric key encryption	ARIA (CBC)	256	SafeDB SDK	Used when storing security policy or cryptographic key encryption with One Day Key	Using random bit generator
TTAK.KO-12.0274	Password Based Key Derivation Functions	HMAC-SHA-2	256	SafeDB Policy Server	For encryption when storing cryptographic key such as Master Key	Entered by the administrator
ISO/IEC 9797-2	Message authentication code	HMAC-SHA-2	256	SafeDB SDK	For examining log forging/corruption with Log Key	Using random bit generator
ISO/IEC 18033-2	Public key encryption	RSAES	2048	SafeDB Policy Server	For encryption when storing Master Key / Log Key	Using RSA key generation algorithm
ISO/IEC 18033-2	Public key encryption	RSAES	2048	TOE inter-module communication	For Session Key encryption	Using RSA key generation algorithm
TTAS.KO-12.0004	Symmetric key encryption	SEED (CBC)	128	TOE inter-module communication	Used when encrypting TOE inter-module communication with Session Key	Using random bit generator
KS X 1213 ISO/IEC	Symmetric key	ARIA (CBC)	256	SafeDB Policy	Configuration file encryption,	Using random bit generator

9797-2	encryption Message authenticati on code	HMAC -SHA- 2		Server, etc.	integrity	

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 encryption policies such as cryptographic key generation on SafeDB Policy Server, and deploys cryptographic keys and security policies to SafeDB Agent. SafeDB Agent receives cryptographic keys and security policies, and encrypts and stores them. When a request for service ID validation is made in SafeDB SDK installed in the user application system, etc. to check the user data encryption privilege, SafeDB Agent performs service ID validation and deploys cryptographic keys and encryption policies to SafeDB SDK. User data encryption/decryption is performed by calling an encryption interface in SafeDB SDK.

- Plug-In type

The authorized administrator establishes encryption policies such as cryptographic key generation on SafeDB Policy Server, deploys cryptographic keys and security policies to SafeDB Agent. SafeDB Agent receives cryptographic keys and security policies, and encrypts and stores them. When a request for service ID validation is made in SafeDB Plug-In installed in the DBMS to check the user data encryption privilege, SafeDB Agent performs service ID validation and deploys cryptographic keys and encryption policies 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)

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. Console Manager can perform the console security management after undergoing the identification and authentication of SafeDB Policy Server Master Console.
The administrator shall enter the following identification and authentication data through the screen interface (satisfying FIA_UID.2, FIA_UAU.2).

- Administrator ID
- Administrator password

If unsuccessful authentication attempts are made for more than a defined number of times (fixed value of 5 times), the account of the relevant ID is locked (fixed value of 5 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, which requires a probabilistic analysis.

The TOE provides mechanisms to verify that administrator passwords meet the following defined quality metrics upon password registration or change (FIA_SOS.1).
a) Allowable characters

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

b) Minimum/Maximum length

- 9 ~ 16 digits

c) Combination rules

- A combination of 3 or more among English alphabets/numbers/special characters
- Excluded if a password has three or more successive letters (numbers)

- Excluded if a password was used the last five times
- 3 special characters on the keyboard (',",;) are excluded

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

- 0 ~ 999 days (the interval is defined by the authorized administrator)

The TOE prevents the reuse of authentication data as follows (FIA_UAU.4):

- The TOE authenticates the administrator through password-based cryptographic authentication.
- The TOE stores and examines the number of unsuccessful attempts and authentication time in the DB upon every authentication.
- The TOE stores and inspects unique session ID in password-based password authentication to prevent reuse of authentication data.

The TSF masks passwords with "*" in SafeDB Manager for users and does not display input characters in the Manager 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 a 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 cryptographic communication between components.

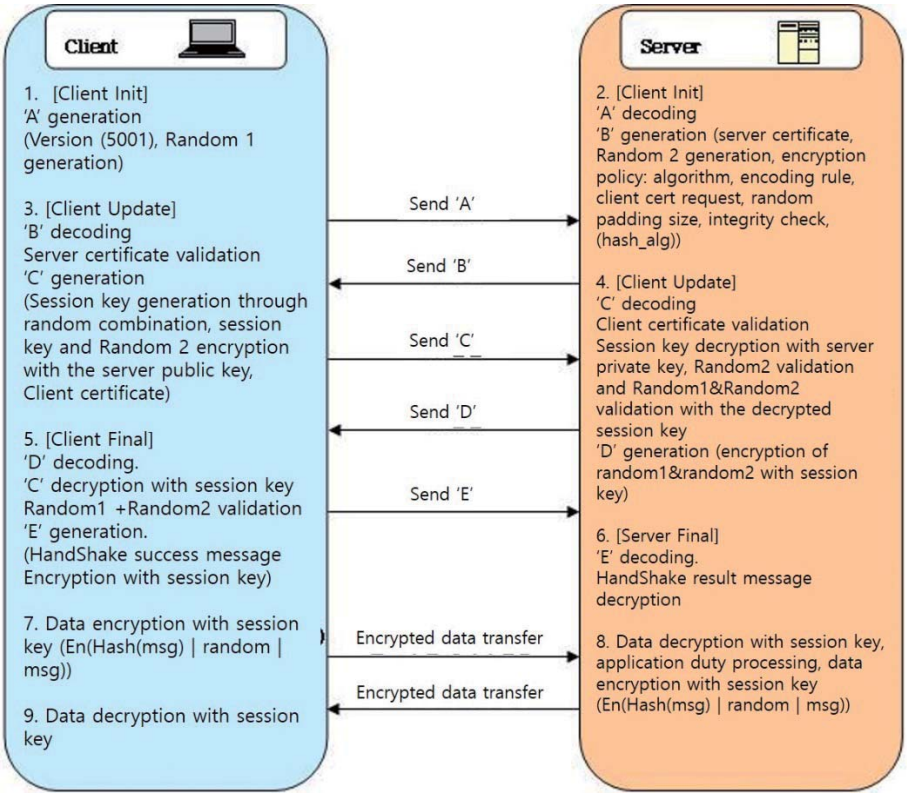
The following table describes TOE internal mutual authentication method and when the mutual authentication runs:

[Table 42] 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
SafeDB SDK	SafeDB Agent	HandShake encryption method	Upon the request of user ID validation
SafeDB	SafeDB	HandShake encryption	Upon the request of user ID

Plug-In	Agent	method	validation
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 43] Encrypted communication procedure

Step	Performance	Behavior
1	Client	Generate Random by using "random bit generator" in *[Table 19] Random bit generator*
	Communication	Send Version + Random1 to the server. Send 'A' in *HandShake encryption method*
2	Server	Generate Random2 by using "random bit generator" in *[Table 19] Random bit generator*
	Communication	The server sends the server certificate, Random2 and encryption policy to the client. Send 'B' in *HandShake encryption method*
3	Client	Validate the validity of the server certificate
	Client	Generate a session key by using "random bit generator" in *[Table 19] Random bit generator*
	Client	Encrypt Random2 with the session key

	Client	Encrypt the session key with a public key
	Communication	Send encrypted Random2 + session key encrypted with the public key + client certificate to the server. Send 'C' in *HandShake encryption method*
4	Server	Validate 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	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	Validate Random1+Random2
	Client	Encrypt HandShake result message with the session key
	Communication	Send the encrypted result message. Send 'E' in *HandShake encryption method*
6	Server	Decrypt encrypted result message with the session key
7	Client/ Server	Encrypt the data that the user wants with the session key. Data encryption format is En(Hash(msg)|random | msg). Msg: data to be encrypted Random: Generate Random in the defined bytes (HashDRBG) and do padding. To prevent the same ciphertext from being generated every time the same msg is encrypted. Hash(msg): Generate Hash value of msg through the set hash algorithm (Sha256) To validate the message integrity En: Encrypt with the set symmetric key algorithm (ARIA-CBC, etc.)
	Communication	Send the data encrypted with the session key. Data transfer in HandShake encryption method
8,9	Server/ Client	Decrypt 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 behavior

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

[Table 44] List of security functions behavior of administrator

Administrator Type	Classification	Security Function	Ability			
			Determine the behavior	disable	enable	Modify the behavior
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	Monitoring	SafeDB Policy Server CPU/memory use	O	-	-	O
		SafeDB Agent CPU/memory use	O	-	-	O
	Additional management	Cryptographic operation status	O	-	-	-
		Audit data review	O	-	-	-
Console Manager	Initialization	TOE initialization setting	O	-	O	-
	Integrity	Configuration	O	-	O	-

	validation	integrity validation file generation				

Administrators are classified into Security Manager (Policy Manager, General Manager) and Console Manager, and performs the security management such as security policy establishment according to their roles in web security management and Master Console.

Application service users request service ID validation in SafeDB SDK and SafeDB Plug-In. Then, they can use the function of cryptographic operation if service ID validation is successful in SafeDB Agent.

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 45] List of TSF data and management ability

Administrator Type	TSF Data	Ability				
		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				-
Console Manager	Security policy					O
	Cryptographic key (Master Key, Log Key, account	-	-	-	O	-

	information encryption key, etc.)					
	Audit data	-	-	-	-	O
	Set value	-	O	-	-	-

Application notes: Console 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 register and modify user's authentication data – password. It also provides the function that enables Policy Manager to change the password when he/she accesses SafeDB Manager for the first time in the process of installing the TSF.
Only the authorized administrator can register and change a password through the screen interface provided by SafeDB Manager. In addition, it provides the function that enables Console Manager to set up a password upon the initial access to Master Console of SafeDB Policy Server and Agent.

Password is encrypted with SHA256 algorithm, including which the security policy is once again encrypted with ARIA algorithm and stored. Refer to FIA_SOS.1 for password lengths and combination rules selected when registering or changing password.

SFR to be satisfied: FMT_PWD.1(Extended)

6.6. Protection of the TSF (FPT)

6.6.1. Basic protection of stored TSF data

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

[Table 46] Protection method in storing cryptographic key and key 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
	Log Key, iv	ARIA 256 CBC	Password-based derivation key	File	1st encryption
	Log Key	RSA 2048	Private certificate public key	File	2nd encryption
SafeDB Agent	User data encryption key (DEK)	ARIA 256 CBC	One Day Key	Memory	Encryption
	One Day Key, iv	ARIA 256 CBC	Derivation key	Memory	Encryption
SafeDB SDK SafeDB Plug-In	User data encryption key (DEK)	ARIA 256 CBC	One Day Key	Memory	Encryption
	One Day Key, iv	ARIA 256 CBC	Derivation key	Memory	Encryption

[Table 47] List of security policy, account information encryption

Classification	Cryptographic Key	Cryptographic 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		Secure hash	SHA-2 (256)	Administrator and service ID password	Policy DB
SafeDB Policy Server	Generated with random bit generator	Symmetric key encryption	ARIA (CBC)	Policy DB account information encryption	File
SafeDB Agent	One Day Key	Symmetric key encryption	ARIA (CBC)	All security policy information	Memory
SafeDB SDK SafeDB Plug-In	One Day Key	Symmetric key encryption	ARIA (CBC)	DEK among security policies	

[Table 48] Configuration file encryption and integrity check algorithm and key

Classification	Type	Algorithm	List of Standards	Key
SafeDB Policy Server	Configuration file encryption	ARIA(CBC)	KS X 1213	256 bit key generated by using "random bit generator" in *[Table 19] Random bit generator*
	Integrity check	HMAC-SHA-2	ISO/IEC 9797-2	
SafeDB Agent	Configuration file encryption	ARIA(CBC)	KS X 1213-1	256 bit key generated by using "random bit generator" in *[Table 19] Random bit generator*
	Integrity check	HMAC-SHA-2	ISO/IEC 9797-2	
SafeDB SDK	Configuration file encryption	ARIA(CBC)	KS X 1213-1	256 bit key generated by using "random bit generator" in *[Table 19] Random bit generator*
	Integrity check	HMAC-SHA-2	ISO/IEC 9797-2	
SafeDB Plug-In	Configuration file encryption	ARIA(CBC)	KS X 1213-1	256 bit key generated by using "random bit generator" in *[Table 19] Random bit generator*
	Integrity check	HMAC-SHA-2	ISO/IEC 9797-2	
SafeDB Manager	Configuration file encryption	ARIA(CBC)	KS X 1213-1	256 bit key generated by using "random bit generator" in *[Table 19] Random bit generator*
	Integrity check	HMAC-SHA-2	ISO/IEC 9797-2	

[Table 49] Configuration file encryption and integrity check key storage

Classification	Algorithm	Cryptographic Key	Application Method	Storage Location
SafeDB Policy Server	RSA 2048	Private certificate public key	Encryption	File
SafeDB Agent	RSA 2048	Private certificate public key	Encryption	File
SafeDB SDK	ARIA(CBC)	Derivation key	Encryption	File
SafeDB Plug-In	ARIA(CBC)	Derivation key	Encryption	File
SafeDB Manager	ARIA(CBC)	Derivation key	Encryption	File

A key that encrypts a configuration file is encrypted with a key encryption key and stored in a file. In case of SafeDB Policy Server and Agent, a configuration file encryption key is encrypted with a public key by using a private certificate as KEK, and decrypts it with a private key. In case of SafeDB SDK and Plug-In, Manager, SafeDB Policy Server and Agent perform the configuration file encryption to generate an encryption key; generates KEK by using a derivation key algorithm; and store it in a file by encrypting the configuration file encryption key. SafeDB SDK, Plug-In and Manager generate KEK by suing a derivation key algorithm same as SafeDB Policy Server and Agent; and decrypt a configuration file encryption key and use it to decrypt the configuration file.

Furthermore, configuration files and execution files that do not include TOE configuration 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)

6.6.2. External entity test

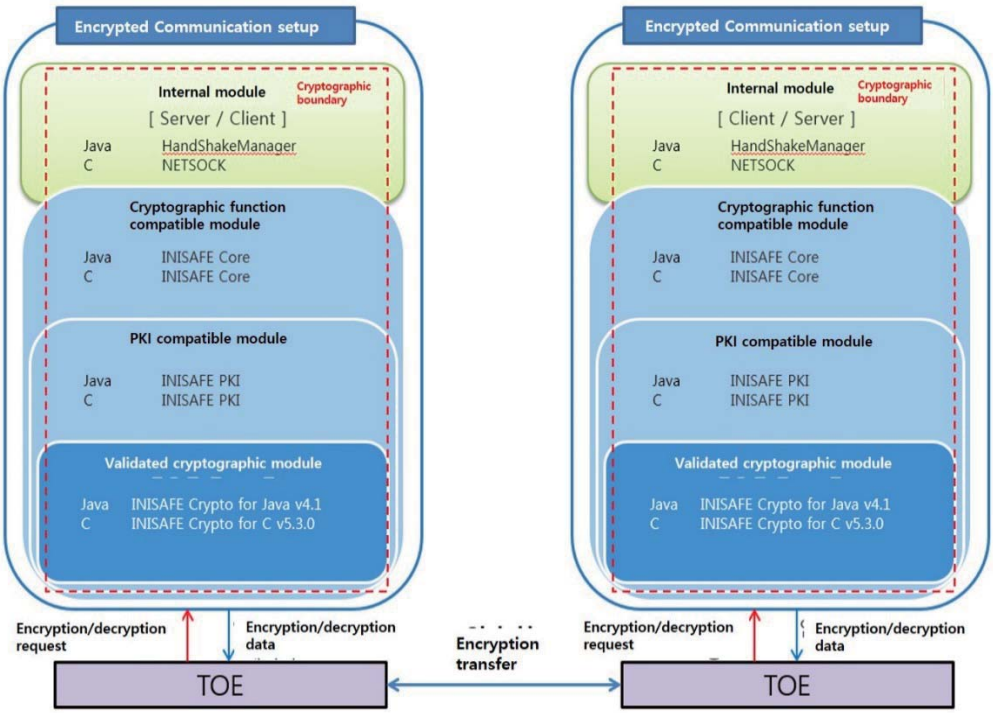
The TOE performs a normal operation test on external entities (mail server, DBMS) upon the initial startup. SMTP access is tested by using mail server information established by the administrator in order to determine the normal operation of the mail server. An access test is carried out to determine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

SFR to be satisfied: FPT_TEE.1

6.6.3. Basic internal TSF data transfer protection

Cryptographic 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 a KCMVP-validated cryptographic module and includes server and client functions in a single internal module. Therefore, it is possible to compose 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 per component of a cryptographic module is indicated below:



(Figure 7) Cryptographic boundary per component

The key management for communication between components as well as communication and mutual authentication between TOE components are performed by means of HandShake encryption method using a KCMVP-validated cryptographic module.
All TOE components have and use the same private certificate (2,048-bit public key) for encrypted communication.

[Table 50] Management of communication encryption key and algorithm used

Classification	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	Using "random bit generator" in *[Table 19] Random bit generator*
	Encryption algorithm	RSA
Transferred data	Encryption algorithm	SEED 128 CBC SHA-256

Refer to 6.4.2 for detailed mechanisms on HandShake encryption method.

SFR to be satisfied: FPT_ITT.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 51] Items subject to TOE self test.

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 51] Items subject to TOE self test

Classification	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 log
SafeDB Agent	Cryptographic module	Self test
	Process	Determine whether it was started normally at the time of start-up and generate audit log 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 52] Items subject to TOE integrity test

Classification	Item	Content (Role)
SafeDB Policy Server	Configuration file	TOE configuration file
	Stored TSF executable code	Policy server daemon process
SafeDB Manager	Configuration file	TOE configuration file
SafeDB Agent	Configuration file	TOE configuration file
	Stored TSF executable code	Agent daemon process

SafeDB SDK	Configuration file	TOE configuration file
	Stored TSF executable code	API operation process
SafeDB Plug-In	Configuration file	TOE configuration file
	Stored TSF executable code	Plug-In operation process

TOE configuration file is protected against unauthorized modification as follows:

[Table 53] Protection of TOE configuration file against unauthorized modification

Classification	Configuration file modification method	How to find modification
SafeDB Policy Server	The administrator executes Master Console function in SafeDB Policy Server, decrypts the configuration file and modifies the content. After the modification, he/she executes Master Console 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 Agent	The administrator executes Master Console function in SafeDB Agent, decrypts the configuration file and modifies the content. After the modification, he/she executes Master Console 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 SDK	The administrator executes Master Console function in SafeDB Agent, decrypts the configuration file and modifies the content. After the modification, he/she executes Master Console 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 Plug-In	The administrator executes Master Console function in SafeDB Agent, decrypts the configuration file and modifies the content. After the modification, he/she executes Master Console 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 Master Console function in SafeDB Policy Server, decrypts the configuration file and modifies the content. After the modification, he/she executes Master Console 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 54] 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
		On a periodic basis	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
		On a periodic basis	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
		On a periodic basis	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
		On a periodic basis	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
		On a	Send an email to the registered

		periodic basis	administrator

[Table 55] 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
		On a periodic basis	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
		On a periodic basis	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
		On a periodic basis	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
		On a periodic basis	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
		On a periodic basis	Send an email to the registered administrator
SafeDB SDK	Configuration file	Upon initial start-up	Transfer it to SafeDB Agent; send an email to the registered administrator; and stop the start-up
		On a periodic basis	Transfer it to SafeDB Agent and send an email to the registered administrator
	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

		On a periodic basis	Transfer it to SafeDB Agent and send an email to the registered administrator
SafeDB Plug-In	Configuration file	Upon initial start-up	Transfer it to SafeDB Agent; send an email to the registered administrator; and stop the start-up
		On a periodic basis	Transfer it to SafeDB Agent and send an email to the registered administrator
	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
		On a periodic basis	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.

In case 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