Alpha DBGuard V2.1 Security Target

V1.3

< Revision History >

Version	Date	Content
1.0	2025.02.05	Initial version created
1.1	2025.07.18	Added missing SFRs and reflected TOE name and component modifications
1.2	2025.09.03	Modified SFRs and updated summary content accordingly
1.3	2025.09.19	Reflection of ITSCC review opinions

< 목 차 >

1 ST Introduction

1.1 ST Reference

[Table 1] ST Reference

Classification	Content
Title	Alpha DBGuard V2.1 Security Target V1.3
ST Version	V1.3
Author	AlphaBit Security Technology Research Institute Co., Ltd.
Publication Date	2025.09.19
Evaluation Criteria	Common Criteria for Information Technology Security Evaluation (Ministry of Science, ICT and Future Planning Notification No. 2013-51)
Common Criteria Version	Common Criteria for Information Technology Security Evaluation, CC:2022 Revision 1, CCMB-2022-11-001 ~ CCMB-2022-11-005, 2022.11 Errata and Interpretation for CC:2022 (Release 1) and CEM:2022 (Release 1), Version 1.1, CCMB-2024-002, 2024.07
Evaluation Assurance Level	EAL1+ (ATE_FUN.1)
Keywords	Database, Encryption

1.2 TOE Reference

[Table 2] TOE Reference

Classification		Content
TOE		Alpha DBGuard V2.1
Version		V2.1.0.3
Components	API module	Alpha DBGuard API V2.1.0.3 Distribution filename: Alpha_DBGuard_API_V2.1.0.3.tgz
	Management Server	Alpha DBGuard Manager V2.1.0.3 Distribution filename: Alpha_DBGuard_Manager_V2.1.0.3.tgz
	Installation/ Operation Manual	Alpha_DBGuard_V2.1_PREOPE_V1.4.pdf
Developer		AlphaBit Security Technology Research Institute Co., Ltd.

1.3 TOE Overview

This Security Targetdefines the security functional requirements and assurance requirements for Alpha DBGuard V2.1, which provides database encryption service to authorized administrators and Applications that call TOE APIs.

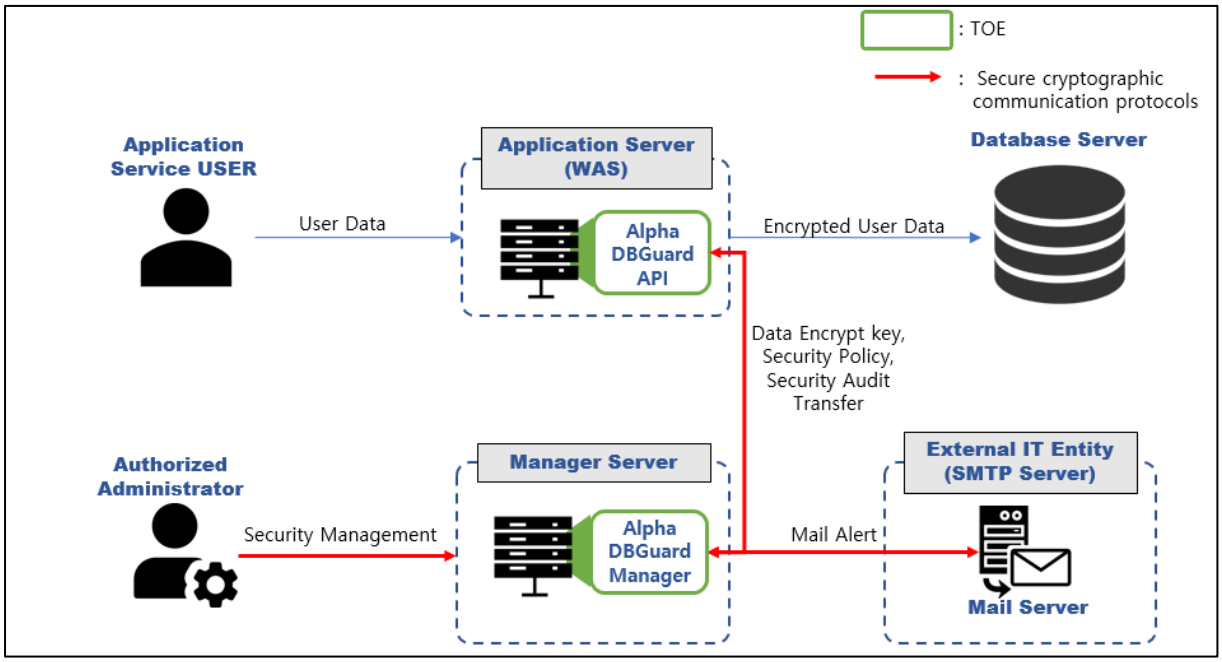
Alpha DBGuard V2.1 (hereinafter referred to as "TOE") performs the function of preventing the unauthorized disclosure of confidential information by encrypting the database (hereinafter referred to as "DB").

The encryption target of the TOE is the DB managed by the database management system in the operational environment of the organization, and this document defines all data before/after encryption and stored in the DB as user data. Part or all of the user data can be the encryption target, depending on the organizational security policies that run the TOE.
The DBMS that controls the DB in the operational environment of the organization is different from the DBMS that is directly used by the TOE to control the TSF data (security policy, audit data, etc.).

The TOE is allowed to encrypt user data according to policies set by authorized administrators to prevent the unauthorized disclosure of information to be protected. The TOE provides various security features: **security audit function** for recording and managing audit data for major auditable events, allowing authorized administrators to securely operate the TOE within the organization's operational environment; **cryptographic support functions** such as cryptographic key management, cryptographic function execution, and random operations for user and TSF data encryption; **user data protection function** that encrypts user data and protects residual information; **identification and authentication** functions such as authentication failure handling and mutual authentication between TOE components; **TSF protection functions** such as data protection for security function and role definition, environment settings, and TSF self-test; **TOE access function** for managing authorized administrator access sessions; and **secure path/channel functions** for communication through secure channels.

1.3.1 TOE Operational Environment

The TOE is provided as "API-type operational environment (API Module, Management Server Separate Type)" for its operational environment. Depending on the environment in which the TOE operates, it is configured as (Figure 1).



[Figure 1] API-type operational environment
(API Module, Management Server Separate Type)

Applications installed on the Application Server to provide application services are developed using APIs provided by the API module(Alpha DBGuard API, hereinafter referred to as "API") to use the TOE's cryptographic functions. The API module is installed on the Application Server and performs encryption/decryption of user data according to policies set by authorized administrators. User data entered by application service users is encrypted by the API module installed on the Application Server and transmitted to the Database Server. Encrypted user data transmitted from the Database Server is decrypted by the API module installed on the Application Server and sent to application service users.

Authorized administrators perform encryption/decryption of user data through the management server(Alpha DBGuard Manager, hereinafter referred to as "Manager") according to the scope of encryption required by the organization's security policy. Additionally, authorized administrators access the management server to perform security management. The management server is physically separated from the API module and installed on the Application Server.

1.3.2 Non-TOE Hardware and Software Requirements

The TOE is software that provides database encryption. All hardware on which the TOE is installed is non-TOE. The minimum hardware and software specifications required for the TOE to be installed are as follows:

[Table 3] TOE Requirements

Category		Minimum Specification
Alpha DBGuard Manager	Hardware	CPU: Intel(R) Core(TM) i5-7200U Dual Core 2.5 GHz or higher Memory: 8GB or higher HDD: Space required for TOE installation 512 MB or higher NIC : 100/1000 Ethernet Port x 1EA
	Software	OS : Rocky Linux 9.6 (64bits) Kernel 5.14.0-570.23.1 WAS : SpringBoot V2.7.18 DB : H2DB V2.2.224 JRE : Oracle Java Runtime Environment v1.8.0_202
Alpha DBGuard API	Hardware	CPU: Intel(R) Core(TM) i5-7200U Dual Core 2.5 GHz or higher Memory: 8GB or higher HDD: Space required for TOE installation 512 MB or higher NIC : 100/1000 Ethernet Port x 1EA
	Software	OS : Rocky Linux 9.6 (64bits) Kernel 5.14.0-570.23.1 WAS : Apache Tomcat 9.0.95 JRE : Oracle Java Runtime Environment v1.8.0_202

The IT entities required in the operating environment are as follows:

- SMTP Server used to send alert emails to administrators
- DBMS Server that stores encrypted user data

The minimum system requirements for an authorized administrator to perform the TOE's security management functions are as follows:

Category	Minimum Requirements
Software	Google Chrome 131.0 or higher

Furthermore, the operating environment required by the TOE is as follows:

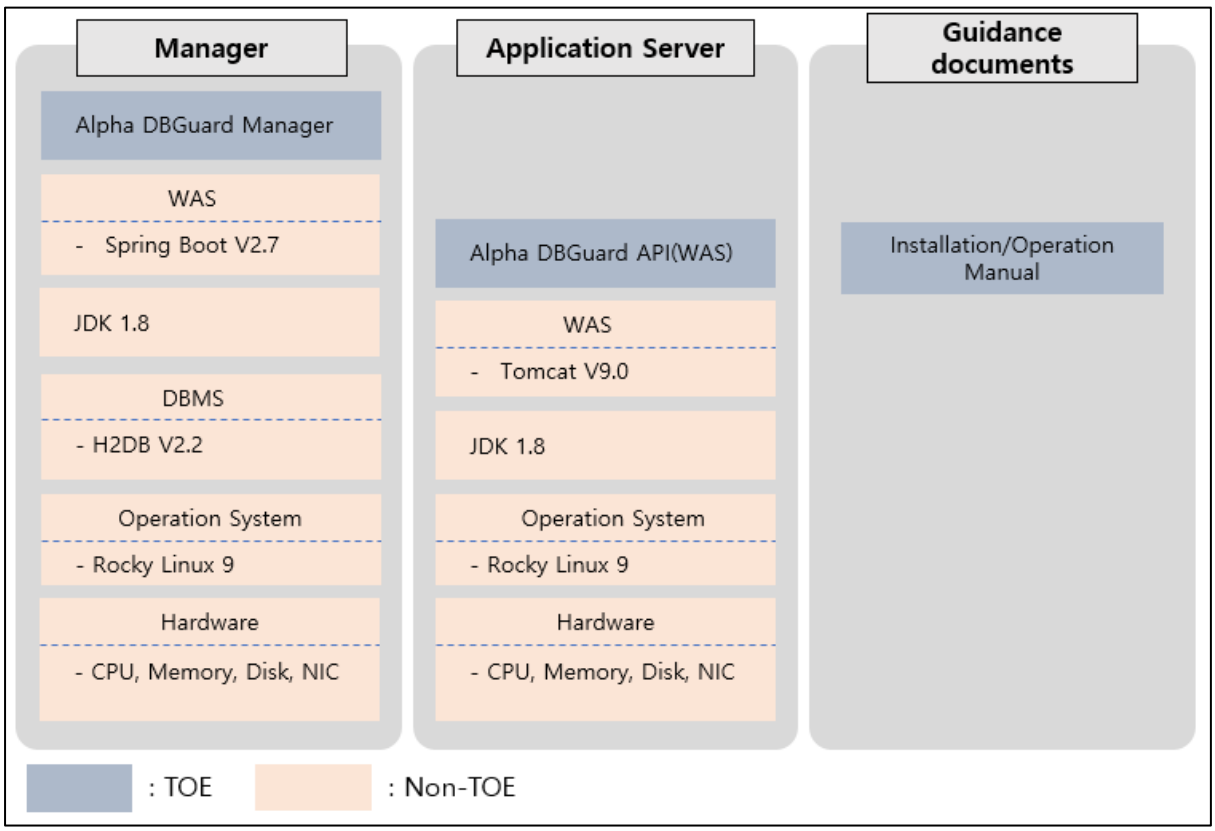
- Spring Boot 2.7.18 for Manager service startup for policy and cryptographic key management
- Apache-Tomcat 9.0.95 for WAS API startup
- JRE 1.8.0_202 for Java-based application startup
- H2DB V2.2.224 DBMS for storing TOE policies, cryptographic keys, and audit data

1.4 TOE Description

This section describes the physical and logical scope and boundaries of the TOE.

1.4.1 Physical Scope of the TOE

The TOE consists of Alpha DBGuard API, Alpha DBGuard Manager, and guidance (Installation/Operation Manual). Hardware, OS, DBMS, WAS, and JDK on which the TOE is installed are not included in the TOE scope.



[Figure 2. TOE Physical Scope]

The TOE includes Alpha DBGuard API, Alpha DBGuard Manager, and Installation/Operation Manual. The operator must prepare JDK 1.8, Spring Boot V2.7, H2DB V2.2, and Tomcat V9.0, which are essential software for TOE operation, before installation

[Table 4] TOE Physical Scope

Classification	Content	Type	Dist Method
TOE	Alpha DBGuard V2.1	-	-

Version		V2.1.0.3	-	-
Components	Alpha DBGuard API	Alpha DBGuard API V2.1.0.3 Distribution filename: Alpha_DBGuard_API_V2.1.0.3.tgz	S/W	CD
	Alpha DBGuard Manager	Alpha DBGuard Manager V2.1.0.3 Distribution filename: Alpha_DBGuard_Manager_V2.1.0.3.tgz	S/W	
	Installation/Operation Manual	Alpha_DBGuard_V2.1_PREOPE_V1.4.pdf	File (PDF)	

The 3rd party software included in the TOE is as follows:

- rt.jar for performing encrypted communication between TOE components and between the TOE and external IT entities (SMTP server)
- AlphaCrypto V1.0 for encryption and decryption of TSF data and user data

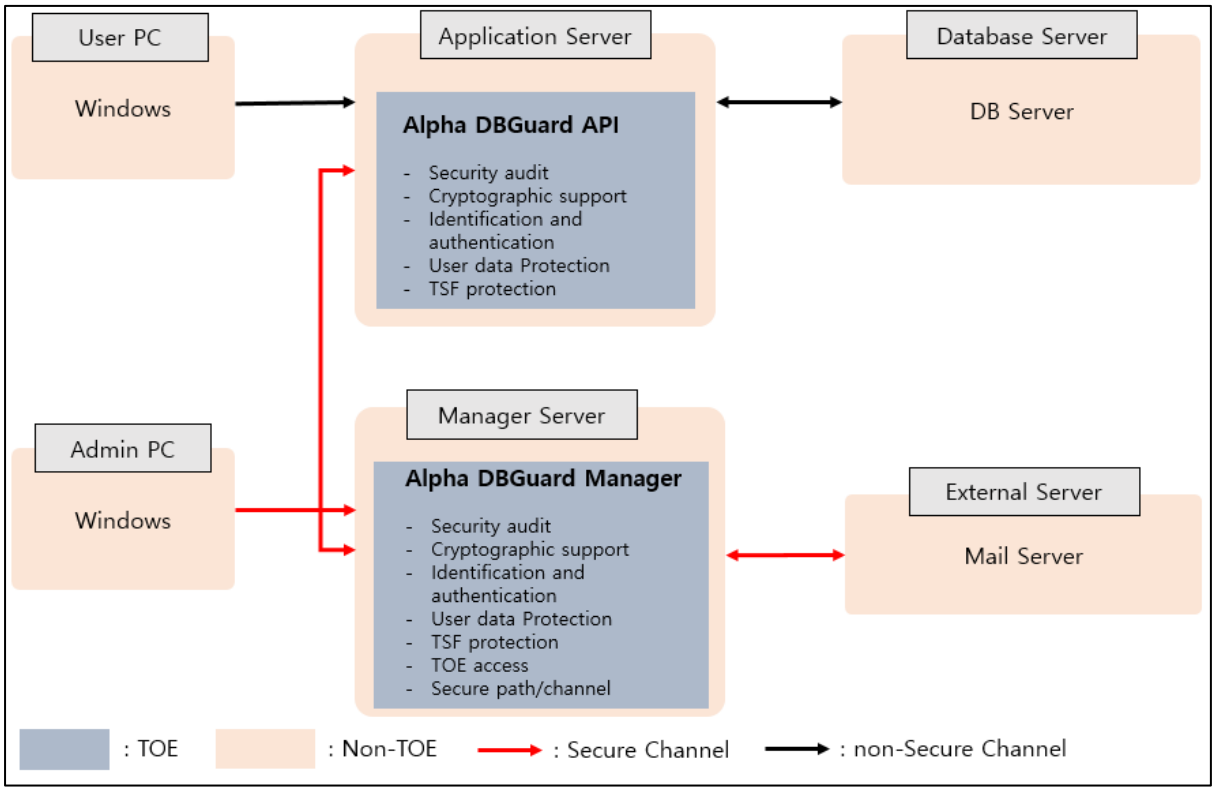
Among the functions provided, the cryptographic function mandatorily uses a cryptographic module whose safety and implementation suitability have been confirmed through the Cryptographic Module Validation Program (KCMVP).

The TOE uses the following validated cryptographic module:

[Table 5] Validated Cryptographic Module

Developer	AlphaBit Co., Ltd.
Cryptographic Module Name	AlphaCrypto V1.0
Validation Number	CM-265-2030.3
Validation Date	2025.03.07
Expiration Date	2030.03.07

1.4.2 Logical Scope of the TOE



[Figure 3. TOE Logical Scope]

The explanation of each module's security functions in the logical scope is as follows:

● Security Audit (Target TOE: Manager, API)

The security audit function consists of audit record generation, audit record review, security violation analysis and response, and audit record protection. Audit records for TOE startup and shutdown, and security-related configuration activities are generated, and generated audit records are protected from unauthorized deletion by OE.DBMS. Only authenticated administrators can view audit data and selectively review audit data. In addition, the TOE analyzes potential violations, such as continuous authentication failures by administrators, sends notifications to authenticated administrators via email, and generates audit records. Furthermore, if the audit storage space exceeds the threshold (default 90%, 60~99%), it sends a notification to authenticated administrators via email, and if the audit evidence storage space becomes saturated (Table Space reaches 99%), subsequent log records are saved after deleting the oldest logs.

● Cryptographic Support (Target TOE: Manager, API)

The TOE performs functions such as cryptographic key management, cryptographic function execution, and random number generation. The algorithms used at this time utilize the validated cryptographic module AlphaCrypto V1.0. To generate cryptographic keys and initial vectors, Hash_DRBG-based random numbers are generated using the validated cryptographic module. For

data encryption, ciphertext is generated through the ARIA block cipher and SHA-256 hash algorithm provided within the validated cryptographic module. Security policies and user data cryptographic key distribution from Manager to API are securely transmitted through a self-defined mutual authentication and secure channel configuration protocol using RSA-2048

● **User data protection (Target TOE: API)**

To protect user data stored within the protected DBMS, the target user data is encrypted and stored through the validated cryptographic module. Encryption/decryption is performed with an ARIA block cipher algorithm that supports 128, 192, 256-bit cryptographic keys, according to the security policy defined by the authorized administrator. For one-way encryption, it is performed via a 256-bit SHA hash algorithm. It provides functions to encrypt and decrypt user data by column, and prevents the generation of identical ciphertexts for identical data when encrypting user data.

● **Identification and authentication (Target TOE: Manager, API)**

The TOE's identification and authentication are performed based on ID and Password. All TOE management functions cannot be used until user authentication is complete. During user authentication, password input prevents exposure by displaying only masking characters, and does not provide reasons for authentication failure. If continuous authentication failures occur, account lockout is performed according to the set value to prevent unauthorized attacks. Password combination rules for authentication are a minimum of 9 characters and a maximum of 20 characters, and must include at least four combinations of uppercase letters/lowercase letters/numbers/special characters. More than three identical consecutive characters cannot be used, and account information should not be included. To ensure communication security between TOE components, Manager and API exchange cryptographic keys for secure communication after mutual authentication using RSA-2048 public key pairs. Subsequently, according to API requests, the Manager encrypts and transmits security policies with ARIA-256. Mutual authentication uses a self-implemented method based on public key cryptography.

● **Security Management (Target TOE: Manager)**

- The TOE provides only a single administrator with administrator privileges and can perform security management only through the Manager after identification and authentication.
- The provided security management functions and managed TSF data are as follows:

a) User data encryption related settings: User data encryption keys and policies

b) Administrator account information: Password, allowed access IP, email address

c) External IT entity settings: SMTP server authentication information

d) TOE operational settings: Audit record storage threshold, account lockout time upon administrator authentication failure, API public key generation, Manager integrity verification

performance
e) Audit record inquiry: Audit records of audit events occurring in the TOE

● Protection of the TSF (Target TOE: Manager, API)

The TOE encrypts and stores TSF data for TOE operation using the symmetric encryption algorithm ARIA-256-CBC of the validated cryptographic module to protect it.

TSF data encryption keys, user data encryption keys, and private keys for mutual authentication are encrypted using KEK derived from PBKDF2.

TOE configuration information and integrity original hash values are encrypted using TSF data encryption keys.

Administrator passwords are stored after being hashed with SHA-256, including salt generated using the validated cryptographic module's random number generator, and an iteration count of 1000.

For data transmission between TOE components, a TLS V1.2 based TLS_ECDHE_RSA_AES_256_GCM_SHA384 cryptographic communication channel is created, and then self-implemented mutual authentication and secure channel establishment are performed for data transmission.

To maintain the secure state of the validated cryptographic module's random number generator, if the validated cryptographic module stops due to a severe error from a health test failure, the cryptographic module is re-initialized once. If the validated cryptographic module repeatedly enters a severe error state, it is considered a self-test failure, and corresponding actions are taken.

The Manager performs self-tests and integrity verification periodically (every hour) during startup and operation. Additionally, upon administrator request, it performs integrity verification. If self-test or integrity verification fails, the TOE operation is terminated, and an alert email is sent to the administrator. The API performs integrity checks when calling a module for user data encryption. If integrity verification fails, the API module operation is terminated.

● TOE Access (Target TOE: Manager)

The TOE provides only a single administrator and a single allowed access IP for administrator information for security management, allowing access to the Manager only from registered IPs. If an administrator performs a duplicate login, the existing session is terminated, and an audit record is generated. If there is no administrator activity for a certain period (10 minutes), the session is terminated, and an audit record is generated.

● Secure Path/Channel (Target TOE: Manager)

When a security event occurs and an email is sent via SMTP, the TOE transmits it through a secure channel based on TLS V1.2 TLS_ECDHE_RSA_AES_256_GCM_SHA384.

1.5 Terms and Definitions

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

Agent Type1

Antivirus products, Software-Based Security USB products, Host Data Loss Prevention products, etc.

- The endpoint on which the agent is located is typically a PC with Windows® operating system accessible to employees within the organization, and if the agent is compromised, data present on the user's host can be compromised and leaked, requiring strict security requirements in terms of confidentiality, integrity, and availability.

Agent Type2

Network Access Control products, Patch Management Systems, etc.

- The endpoint on which the agent is located is typically a PC with Windows® operating system accessible to employees in the organization, and if the agent is compromised, it is unlikely that data present on the user's host will be corrupted or leaked, but it can cause problems in using the resources provided by the organization, requiring security requirements in terms of confidentiality, integrity.

Agent Type3

Database Access Control products, Access Control in Operating System(Server) products, Enterprise security management products, etc.

- Since the endpoint where the agent is located is generally a physically secure environment that can only be accessed by authorized employees of the organization, it corresponds to a product type with a relatively low threat occurrence.

Approved cryptographic algorithm

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

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.

Approved mode of operation

The mode of cryptographic module using approved cryptographic algorithm

Assets

Entities that the owner of the TOE presumably places value upon

Assignment

Specification of an identified parameter in a functional or assurance component

Attack potential

Measure of the effort needed to exploit a vulnerability in a TOE

Note 1 to entry: The effort is expressed as a function of properties related to the attacker (e.g. expertise, resources, and motivation) and properties related to the vulnerability itself (e.g. window of opportunity, time to exposure).

Augmentation

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

Note 1 to entry: In case of a functional package), such an augmentation is considered only in the context of one package and is not considered in the context with other packages or PPs or STs.

Note 2 to entry: In case of an assurance package, augmentation refers to one or more SARs.

Authorized Administrator

Authorized user to securely operate and manage the TOE

Authentication Data

Information used to verify the claimed identity of a user

Authorized User

Entity who may, in accordance with the SFRs, perform an operation on the TOE

Automated recovery

Recovery without the user's intervention

Can/could

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

Column

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

Component

Smallest selectable set of elements on which requirements may be based

Conditioning

The process of increasing the entropy rate per bit by removing the bias from collected noise sources

Critical Security Parameters (CSP)

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

Class

Set of CC families that share a common focus

Client Type

Virtual Private Network products, Wireless LAN Authentication Products, etc.

- The client is an entity installed on the user's host and serves to request communication with the server on behalf of the user.

Database

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

Database Server

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

DBMS (Database Management System)

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

Data Encryption Key (DEK)

Key that encrypts and decrypts the data

Decryption

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

Dependency

Relationship between components such that a PP, ST, functional package or assurance package including a component also includes any other components that are identified as being depended upon or include a rationale as to why they are not

Deterministic Random Bit Generator (DRBG)

It consists of an algorithm that generates a bit string from an initial value called a seed and produces the same bit string when the same seed is input.

Encryption

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

Element

Self-contained description of a security need assigned to SAR or SFR

Endpoint

The point where the TOE components such as agents, clients, etc. are installed and operated without any further sub-interacted entities

Entropy

A measure used to evaluate the unpredictability of data.

A numerical representation of the amount of information contained in data.

It represents disorder or randomness, and the closer it is to a random bit, the higher the entropy.

Entropy rate

The entropy of the data divided by the size of the data, expressed as a value between 0 and 1.

Entropy source

A function or device that combines noise sources, health tests, and conditioning algorithms

External Entity

Human technical system or one of its components interacting with the target of evaluation TOE from outside of the TOE boundary

Evaluation Assurance Level (EAL)

Well-formed package of security assurance requirements representing a point on the pre-defined assurance scale

Note 1 to entry: EALs are defined in CC Part 5.

Family

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

Health test

Implemented within a random bit generator to monitor noise sources in real time.

The health test is not a process for identifying statistical problems with noise sources; rather, it is a method for detecting cases where the collected noise sources do not operate normally due to equipment aging, etc.

※ For detailed information, refer to the health test defined in Section 5.2 of TTAK.KO-12.0306/R1.

Identity

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

Iteration

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

KCMVP, Korea Cryptographic Module Validation Program

A system to validate the security and implementation conformance of cryptographic modules used for the protection of important but not classified information among the data communicated through the information and communication network of the government and public institutions.

Key Encryption Key (KEK)

Key that encrypts and decrypts another cryptographic key

Local access

Connection established through the console port between the administrator and the TOE

Management access

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

Manual recovery

Recovery through an update server, etc. by the user execution or user intervention

Noise Source

Functions or devices that generate non-deterministic data

Object

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

Operation (on a component of the CC)

Modification or repetition of a component. Allowed operations on components are assignment, iteration, refinement and selection

Operation (on an object)

⟨on an object⟩ specific type of action performed by a subject on an object

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

Private Key

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

Protection Profile (PP)

Implementation-independent statement of security needs for a TOE type

Public Key

A cryptographic key which is used in an asymmetric cryptographic algorithm and is associated

with an unique entity (the subject using the public key), it can be disclosed

Public Key (asymmetric) cryptographic algorithm

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

Random bit generator (RBG)

A device or algorithm that outputs a binary sequence that is statistically independent and is not biased. The RBG used for cryptographic application generally generates 0 and 1 bit string, and the sequence 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.

※ The cryptographic random bit generator consists of an entropy source used for seed construction and a deterministic random bit generator.

Recommend/be recommended

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

Refinement

Addition of details to a security component

Role

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

Security Function Policy (SFP)

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

Secret Key

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

Security Target (ST)

Implementation-dependent statement of security requirements for a TOE based on a security problem definition

Security attribute

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

Security Token (HSM)

A hardware device implemented to process key generation, electronic signature generation, etc., within the device in order to safely save and store confidential information.

Seed

The secret value used to initialize the random bit generator

Selection

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

Shall/must

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

Subject

Entity in the TOE that performs operations on objects

SSL (Secure Sockets Layer)

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

Symmetric cryptographic technique

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

Subject

Active entity in the TOE that performs operations on objects

Target of Evaluation (TOE)

Set of software, firmware and/or hardware possibly accompanied by guidance, which is the subject of an evaluation

Threat Agent

Entity that has potential to exercise adverse actions on assets protected by the TOE

TLS (Transport Layer Security)

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

TOE Security Functionality (TSF)

Combined functionality of all hardware, software, and firmware of a TOE that is relied upon for the correct enforcement of the SFRs

TSF Data

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

User

As a human technical system or one of its components interacting with the TOE from outside the TOE boundary, the user in the TOE is an authorized administrator and an authorized end user.

※ The types of users related to the SFR are divided into human users and external IT entities. Human users may further be differentiated as local human users, meaning they interact directly with the TOE via TOE devices (e.g. workstations), or remote human users, meaning they interact indirectly with the TOE through another IT product.

User Data

Data for the user, that does not affect the operation of the TSF

1.6 Writing Rule

This Security Target uses selection, assignment, refinement, and iteration operations in the same way as the Common Criteria.

2 Conformance Claim

2.1 CC Conformance Claim

The Common Criteria and Protection Profile, and Assurance Requirements Package that the Security Target and TOE comply with are as follows.

Classification	Compliance
Common Criteria	Common Criteria for Information Technology Security Evaluation, CC:2022, Revision 1 - Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model, CC:2022 R1 (CCMB-2022-11-001, 2022. 11.) - Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components, CC:2022 R1 (CCMB-2022-11-002, 2022.11.) - Common Criteria for Information Technology Security Evaluation. Part 3: Security Assurance Components, CC:2022 R1 (CCMB-2022-11-003, 2022.11.) - Common Criteria for Information Technology Security Evaluation. Part 4: Framework for the specification of evaluation methods and activities, CC:2022 R1 (CCMB-2022-11-004, 2022.11.) - Common Criteria for Information Technology Security Evaluation. Part 5: Pre-defined packages of security requirements, CC:2022 R1 (CCMB-2022-11-005, 2022.11. - Errata and Interpretation for CC:2022 (Release 1) and CEM:2022 (Release 1), Version 1.1, (CCMB-2024-002, 2024.07)
Conformance claim part 2	Extended: FDP_UDE.1, FIA_IMA.1, FMT_PWD.1, FPT_PST.1
Conformance claim part 3	Conformant
Conformance Type	Strict Conformance
Protection Profile	Korean National Protection Profile for Database Encryption V3.1
Assurance Requirements Package	EAL1 *augmented* (ATE_FUN.1)

2.2 PP Conformance Claim

The Protection Profile that this Security Target complies with is as follows:

Classification	Compliance
Protection Profile	Korean National Protection Profile for Database Encryption V3.1

	(KECS-PP-1350-2025)
Conformance Type	Strict Conformance

2.3 Package Conformance Claim

This Security Target claims no conformance to any package.

2.4 Conformance Claim Rationale

This Security Target has strictly complied with the "Korean National Protection Profile for Database Encryption V3.1" in terms of security problem definition, security objectives, and security requirements

[Rationale]
The rationale for the added security objectives for the operational environment, according to the SFRs identified in the "Korean National Protection Profile for Database Encryption V3.1", is as follows:

- OE.AUDIT_DATA_PROTECTION : Added to satisfy FAU_STG.2 requirements among SFRs identified in PP
- OE.TIME_STAMP: Added to satisfy FPT_STM.1 requirements among SFRs identified in PP
- OE.MANAGEMENT_ACCESS: Added to satisfy FPT_TRP.1 requirements among SFRs identified in PP

2.5 Compliance Method

2.5.1 Reference to Evaluation Methods/Activities

The "EAL1+" package, which this Security Target complies with, requires the use of the evaluation methods/activities defined in `<6.2. Security Assurance Requirements>`.
The "Korean National Protection Profile for Database Encryption V3.1", which this Security Target complies with, requires the use of the evaluation methods/activities defined in `<Korean National Protection Profile for Database Encryption V3.1 Supporting Document>`

3 Security problem definition

3.1 Assets

The basic assets protected by Database Encryption are as follows.

- Database managed by DBMS in the organization's operational environment.
- Important data related to the TOE itself and TOE operation (e.g. TSF data)

3.2 Threats

Threat agents are IT entities and human users that cause harm to assets through unauthorized access or abnormal methods, and can generate various threats as follows. At this time, threat agents for the TOE have a basic level of expertise, resources, and motivation.

3.2.1 Unauthorized access

T.SESSION_HIJACK

Threat agents can seize administrator privileges by accessing administrator screens left logged in or by using administrator sessions that have not been terminated after logout.

T.RETRY_AUTH_ATTEMPT

Threat agents can access the TOE by impersonating an authorized user after successfully authenticating using information obtained through continuous authentication attempts

T.IMPERSONATION

Threat agents can access the TOE by impersonating authorized users, TOE components, etc.

T.REPLAY

Threat agents can obtain and copy authentication information and reuse it to access the TOE.

T.WEAK_PASSWORD

Threat agents can obtain poorly managed passwords, such as using default password values, and impersonate authorized users to access the TOE; if low-level password rules are applied, they can impersonate authorized users to access the TOE.

3.2.2 Information leak

T.UNAUTHORIZED_INFO_LEAK

Threat agents can leak important user information stored in the database through unauthorized

means.

T. STORED_DATA_LEAKAGE

Threat agents can leak important data (e.g., cryptographic keys, TOE settings) stored within the TOE or in external entities (e.g., DBMS) interacting with the TOE through unauthorized means.

T.TRANSMISSION_DATA_DAMAGE

Threat agents can unauthorizedly expose or modify data transmitted between TOE components and external IT entities

T.WEAK_CRYPTO_PROTOCOLS

Threat agents can infer cryptographic key information or ascertain the contents of communication ciphertexts by analyzing traffic that uses weak cryptographic communication protocols or low cryptographic strength.

3.2.3 TOE functionality compromise

T.TSF_COMPROMISE

Threat agents can compromise the TSF through unauthorized access, causing TOE functions to malfunction or rendering them inoperable.

3.3 Organizational security policy

P.AUDIT

Security-related events must be recorded and maintained to trace accountability for security-related actions, and recorded data must be reviewed. Additionally, available space on the disk for audit data storage must be regularly checked to prevent audit data loss, and stored audit data must be protected from unauthorized modification and deletion.

P.SECURE_OPERATION

Management means shall be provided so that the administrator can securely set up the TOE to comply with the organizational security policy and operate it correctly according to the TOE operation manual.

P.CRYPTO_STRENGTH

The organization must apply encryption measures to important data storage and transmission sections, such as passwords for user authentication, and must use secure cryptographic algorithms.

3.4 Assumptions

A.PHYSICAL_CONTROL

The place where the TOE is installed and operated shall be equipped with access control and protection facilities so that only authorized administrators can access it.

A.TRUSTED_ADMIN

The authorized administrator of the TOE is non-malicious, has been appropriately trained for the TOE management functions, and accurately fulfills their duties in accordance with administrator guidelines.

A.SECURE_DEVELOPMENT

Developers who use the TOE to link encryption functions to applications or DBMS must comply with the requirements of the manual provided with the TOE to ensure that the security functions of the TOE are safely applied.

A.OPERATION_SYSTEM_REINFORCEMENT

The TOE performs reinforcement on the latest vulnerabilities of the operating system on which it is installed and operated to ensure the reliability and security of the operating system.

A.TIME_STAMP

The TOE must receive reliable timestamps provided by the operating environment to accurately record security-related events.

4 Security objectives

The following security objectives for the operational environment are objectives that must be addressed by technical/procedural means supported by the operational environment to ensure that the TOE accurately provides security functionality.

4.1 Security Objectives for the Operational Environment

OE.LOG_BACKUP

The authorized administrator of the TOE shall periodically check the spare space of audit data storage in case of audit data loss, and perform audit data backup (e.g., external log server, separate storage device) to prevent audit records from being exhausted.

OE.PHYSICAL_CONTROL

The place where the TOE is installed and operated shall be equipped with access control and

protection facilities so that only authorized administrators can access it.

OE.TRUSTED_ADMIN

The authorized administrator of the TOE is non-malicious, has been appropriately trained for TOE management functions, and must accurately fulfill their duties in accordance with administrator guidelines.

OE.SECURE_DEVELOPMENT

Developers who use the TOE to link encryption functions to applications or DBMS must comply with the requirements of the manual provided with the TOE to ensure that the security functions of the TOE are safely applied.

OE.OPERATION_SYSTEM_REINFORCEMENT

The TOE must perform reinforcement on the latest vulnerabilities of the operating system on which it is installed and operated to ensure the reliability and security of the operating system.

OE.AUDIT_DATA_PROTECTION

Audit records stored as audit evidence, such as in a DBMS interacting with the TOE, must be protected from unauthorized deletion or modification.

OE.TIME_STAMP

The TOE must receive reliable timestamps provided by the operating environment to accurately record security-related events.

OE.MANAGEMENT_ACCESS

This ensures that when an authorized administrator accesses the management server via a web browser, all information transmitted is protected through a secure path/channel.

4.2 Security Objectives Rationale

4.2.1 Operational Environment Security Objectives Rationale

[Table 6] Operational Environment Security Objectives Rationale

	OE.LOG_BACKUP	OE.PHYSICAL_CONTROL	OE.TRUSTED_ADMIN	OE.SECURE_DEVELOPMENT	OE.OS_REINFORCEMENT	OE.AUDIT_DATA_PROTECTION	OE.TIME_STAMP	OE.MANAGEMENT_ACCESS
P.AUDIT	○					○		
P.SECURE OPERATION			○					

P.CRYPTO_STRENGTH								○
A.PHYSICAL_CONTROL		○						
A.TRUSTED_ADMIN	○		○					
A.SECURE_DEVELOPMENT				○				
A.OPERATION_SYSTEM_REINFORCEMENT					○			
A.TIME_STAMP							○	
A.MANAGEMENT_ACCESS								○

P.AUDIT OE.LOG_BACKUP, OE.AUDIT_DATA_PROTECTION

P.AUDIT is performed by **OE.LOG_BACKUP, OE.AUDIT_DATA_PROTECTION**.

OE.LOG_BACKUP ensures that regular audit data storage space is checked by the administrator as well as the TOE function, and regular log backups or log transmission to an external log server are performed to prevent log records from being lost.

OE.AUDIT_DATA_PROTECTION ensures that audit records stored as audit evidence, such as in a DBMS interacting with the TOE, are protected from unauthorized deletion or modification.

P.SECURE_OPERATION OE.TRUSTED_ADMIN

P.SECURE_OPERATION is performed by **OE.TRUSTED_ADMIN**.

OE.TRUSTED_ADMIN ensures that the administrator operates TOE accurately in accordance with the organizational security policy and operating manual.

P.CRYPTO_STRENGTH OE.MANAGEMENT_ACCESS

P.CRYPTO_STRENGTH is performed by **OE.MANAGEMENT_ACCESS**.

OE.MANAGEMENT_ACCESS ensures that when an authorized administrator accesses the management server via a web browser, all information transmitted is protected through a secure path/channel.

P.PHYSICAL_CONTROL OE.PHYSICAL_CONTROL

P.PHYSICAL_PROTECTION is supported by **OE.PHYSICAL_CONTROL**.

OE.PHYSICAL_CONTROL places the management server in a place with protective facilities, and controls access so that only authorized administrators can access them.

A.TRUSTED_ADMIN OE.TRUSTED_ADMIN, OE.LOG_BACKUP

A.TRUSTED_ADMIN is supported by **OE.TRUSTED_ADMIN, OE.LOG_BACKUP.**

OE.TRUSTED_ADMIN ensures that the administrator is non-malicious, is properly trained in TOE management functions, and performs their duties accurately according to administrator guidelines. **OE.LOG_BACKUP** ensures that the authorized administrator periodically checks the spare space of audit data storage in case of audit data loss, and performs audit data backup (external log server or separate storage device, etc.) to prevent audit data loss.

A.SECURE_DEVELOPMENT	OE.SECURE_DEVELOPMENT

A.SECURE_DEVELOPMENT is supported by **OE.SECURE_DEVELOPMENT.**

OE.SECURE_DEVELOPMENT ensures that developers who use the TOE to link encryption functions to applications or DBMS comply with the requirements of the documentation provided with the TOE so that the security functions of the TOE are safely applied.

A.OPERATION_SYSTEM_RE-INFORCEMENT	OE.OPERATING_SYSTEM_REINFORCEMENT

A.OPERATION_SYSTEM_RE_INFORCEMENT is support by **OE.OPERATING_SYSTEM_REINFORCEMENT.**

OE.OPERATING_SYSTEM_REINFORCEMENT ensures that the TOE performs reinforcement on the latest vulnerabilities of the operating system on which it is installed and operated to ensure the reliability and security of the operating system.

A.AUDIT_DATA_PROTECTION	OE.AUDIT_DATA_PROTECTION

A.AUDIT_DATA_PROTECTION is supported by **OE.AUDIT_DATA_PROTECTION.**

OE.AUDIT_DATA_PROTECTION ensures that audit records stored as audit evidence, such as in a DBMS interacting with the TOE, are protected from unauthorized deletion or modification.

A.TIME_STAMP	OE.TIME_STAMP

A.TIME_STAMP is supported by **OE.TIME_STAMP.**

OE.TIME_STAMP ensures that the TOE receives reliable timestamps provided by the operating environment to accurately record security-related events.

A.MANAGEMENT_ACCESS	OE.MANAGEMENT_ACCESS

A.MANAGEMENT_ACCESS is supported **OE.MANAGEMENT_ACCESS.**

OE.MANAGEMENT_ACCESS ensures that when an authorized administrator accesses the management server via a web browser, all information transmitted is protected through a secure path/channel.

5 Extended components definition

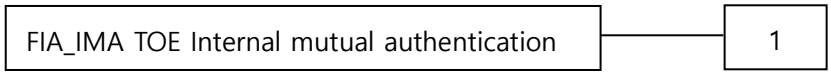
5.1 Identification and authentication (FIA)

5.1.1 TOE Internal mutual authentication

Family Behavior

This family defines requirements for providing mutual authentication between TOE components in the process of user identification and authentication.

Component leveling



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

Management: FIA_IMA.1

There are no management activities foreseen.

Audit: FIA_IMA.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: Success and failure of mutual authentication

FIA_IMA.1	TOE Internal mutual authentication.	
	Component relationships	
	Hierarchical to	No other components.
	Dependencies	No dependencies.
FIA_IMA1.1	The TSF shall perform mutual authentication between [assignment: different parts of TOE] using the [assignment: authentication protocol] that meets the following [assignment: list of standards]	

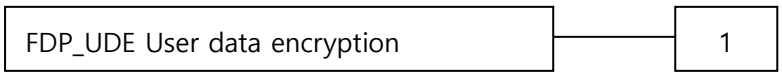
5.2 User data protection (FDP)

5.2.1 User data encryption

Family Behavior

This family provides requirements to ensure confidentiality of user data.

Component leveling



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

Management : FDP_UDE.1

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

a) Management of user data encryption/decryption rules

Audit : FDP_UDE.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

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

FDP_UDE.1 User data encryption.

Component relationships

Hierarchical to No other components.

Dependencies FCS_COP.1 Cryptographic operation

FDP_UDE.1.1 TSF shall provide TOE users with the ability to encrypt/decrypt user data according to [assignment: the list of encryption/decryption methods] specified.

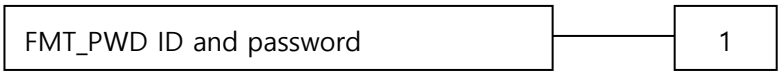
5.3 Security Management (FMT)

5.3.1 ID and password

Family Behavior

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

Component leveling



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

Management: FMT_PWD.1

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

a) Management of ID and password configuration rules.

Audit: FMT_PWD.1
The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: All changes of the password.

FMT_PWD.1 Management of ID and password.

Component relationships

Hierarchical to No other components.

Dependencies FMT_SMF.1 Specification of management functions

FMT_SMR.1 Security roles

FMT_PWD.1.1 The TSF shall restrict the ability to manage the password of [assignment: list of functions] to [assignment: the authorized identified roles].

1. [assignment: password combination rules and/or length]
2. [assignment: other management such as management of special characters unusable for password, etc.

FMT_PWD.1.2 The TSF shall restrict the ability to manage the ID of [assignment: list of functions] to [assignment: the authorized identified roles].

1. [assignment: ID combination rules and/or length]
2. [assignment: other management such as management of special characters unusable for ID, etc.]

FMT_PWD.1.3 The TSF shall provide the capability for [selection, choose one of: setting ID and password when installing, setting password when installing, changing the ID and password when the authorized administrator accesses for the first time, changing the password when the authorized administrator accesses for the first time]

5.4 Protection of the TSF (FPT)

5.4.1 Protection of stored TSF data

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

Component leveling

FPT_PST.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.	
	Component relationships	
	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].

6 Security Requirements

This section describes the security functional requirements and assurance requirements that the TOE must satisfy.

The security functional requirements defined in this Security Target are expressed by selecting relevant security functional components from Common Criteria Part 2 and Section 4 Extended Components Definition.

6.1 Security functional requirement

The security functional requirements defined in this Security Target are expressed by selecting relevant security functional components from CC Part 2 to satisfy the security objectives identified in Section 4. The following [Table 7] summarizes the security functional components used in this Security Target.

[Table 7] Security Functional Requirements

Functional Class	Security Functional Components	
Security Audit (FAU)	FAU_ARP.1	Security alarms
	FAU_GEN.1	Audit data generation
	FAU_SAA.1	Potential violation analysis
	FAU_SAR.1	Audit review
	FAU_SAR.3	Selectable audit review
	FAU_STG.1	Audit data storage location
	FAU_STG.4	Action in case of possible audit data loss
	FAU_STG.5	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.5	Cryptographic key derivation
	FCS_CKM.6	Cryptographic key destruction time and incident
	FCS_COP.1(1)	Cryptographic operation (User data encryption)
	FCS_COP.1(2)	Cryptographic operation (TSF data encryption)
	FCS_COP.1(3)	Cryptographic operation (Hash)

	FCS_COP.1(4)	Cryptographic operation (Digital signature generation)
	FCS_COP.1(5)	Cryptographic operation (Digital signature verification)
	FCS_COP.1(6)	Cryptographic operation (Public key encryption)
	FCS_COP.1(7)	Cryptographic operation (Public key decryption)
	FCS_RBG.1	Random bit generation
	FCS_RBG.3	Random bit generation (Internal seeding – Single source)
User Data Protection (FDP)	FDP_UDE.1(Extended)	User data encryption
	FDP_RIP.1	Subset residual information protection
Identification & Authentication (FIA)	FIA_AFL.1	Authentication failure handling
	FIA_IMA.1(Extended)	TOE Internal mutual authentication
	FIA_SOS.1	Verification of secrets
	FIA_UAU.2	User authentication before any action
	FIA_UAU.4	Single-use authentication mechanisms
	FIA_UAU.7	Protected authentication feedback
	FIA_UID.2	User identification 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)	ID and password management
	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)	Protection of stored TSF data
	FPT_TST.1	TSF self-testing
	FPT_FLS.1	Failure with preservation of secure state
TOE Access (FTA)	FTA_MCS.2	Per user attribute limitation on multiple concurrent sessions
	FTA_SSL3	Management of TSF-initiated sessions
	FTA_TSE.1	TOE session establishment
Secure Path/Channel (FTP)	FTP_ITC.1	Inter-TSF trusted channel

6.1.1 Security Audit

6.1.1.1 FAU_ARP.1 Security alarms

Hierarchical to No other components.
Dependencies FAU_SAA.1 Potential violation analysis

FAU_ARP.1.1 The TSF shall take [[Table 8] Security violation response actions] upon detection of a potential security violation.

[Table 8] Security Violation Response Actions

Security Functional Component	Security Violation	Action
FIA_AFL.1	Administrator account lockout upon administrator continuous authentication attempt failure (default 5 times)	Send E-Mail to authorized administrator
FPT_TST.1	API : upon integrity verification failure	Execution termination (error handling)
	Manager : upon integrity verification failure	Send E-Mail to authorized administrator and execution termination (error handling)
	Manager : upon self-test failure	Send E-Mail to authorized administrator and execution termination (error handling)
FAU_STG.4	If audit data storage exceeds defined capacity (default 90%)	Send E-Mail to authorized administrator
FAU_STG.5	If audit data storage is full	Delete oldest audit data and record, send E-Mail to authorized administrator

6.1.1.2 FAU_GEN.1 Audit data generation

Hierarchical to No other components.
Dependencies FPT_STM.1 Reliable time stamps

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

a) Start-up and shut-down of the audit functions.
b) All auditable events for the *not specified* level of audit.

c) For each audit event type, refer to [Table 9] for functional components included in the Security Target.

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

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP, PP-Module, functional package or ST, [[Table 9] Additional audit information].

[Table 9] Auditable Events

Security Functional Component	Audit Event	Additional Audit Information
FAU_STG.4	Threshold exceeded Response action result when threshold is exceeded	
FAU_STG.5	Audit storage failure Response action result when audit storage fails	
FCS_CKM.1(1)	Cryptographic key generation failure	
FCS_COP.1(1)	Cryptographic operation failure (including cryptographic operation type)	
FDP_UDE.1	Success and failure of user data encryption and decryption	Policy name, encryption/decryption algorithm
FIA_AFL.1	Response action performed and result (success/failure) upon reaching user authentication attempt limit	
FIA_IMA.1	Success and failure of mutual authentication	Authentication entity information
FIA_UAU.2	User login success/failure	
FIA_UAU.4	Authentication failure due to detection of authentication information reuse attempt	
FIA_UID.2	All uses of the administrator identification mechanism, including the provided administrator identity	
FMT_MOF.1	All changes to the [Security Management Function List] specified in FMT.MOF1.1 Excluding audit record view and TOE version information view functions	Changed security attribute data
FMT_MTD.1	User registration/modification/deletion	Changed TSF data value
	All changes to passwords	

	All changes to the 'TSF Data List' specified in FMT_MTD.1.1	
FMT_PWD.1	All changes to passwords	
FPT_TST.1	TOE server self-test execution and result (success, failure)	Failed security function
	Integrity verification performed on TOE components and result (success, failure)	Failed component for integrity check
FTA_MCS.2	Termination of previous session based on limiting concurrent access for single account	
FTA_SSL.5	Termination of interactive session	
FTA_TSE_1	Blocking management terminal access IP	
기타	User logout success	

6.1.1.3 FAU_SAA.1 Potential violation analysis

Hierarchical to 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.
· Self-test and integrity violation audit event of cryptographic module among auditable events of FIA_TST.1.
· Audit storage threshold exceeded audit event among auditable events of FAU_STG.4.
· Audit storage saturation audit event among auditable events of FAU_STG.5.
] known to indicate a potential security violation

b) [None]

6.1.1.4 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.

6.1.1.5 FAU_SAR.3 Selectable audit review

Hierarchical to No other components.

Dependencies FAU_SAR.1 Audit review.

FAU_SAR.3.1 The TSF shall provide the capability to apply [search method in [Table 10]] for audit data based on [selection criteria in [Table 10]].

[Table 10] Audit Data Type and Selection Criteria

Audit Data Type	Selection Criteria (AND)	Search Method
TOE Log	Date and time	Search and sort (descending order based on audit data creation time)
	Log Level	
	Log occurrence type (Manager, API, Administrator)	
	Function Type	
	Result Message	

6.1.1.6 FAU_STG.1 Audit data storage

Hierarchical to No other components.

Dependencies FAU_GEN.1 Audit data generation.
FTP_ITC.1 Inter-TSF trusted channel.

FAU_STG.1.1 The TSF shall be able to store generated audit data on the *[store in DBMS interworking with TOE]*

6.1.1.7 FAU_STG.4 Action in case of possible audit data loss

Hierarchical to No other components.

Dependencies FAU_STG.2 Protected audit data storage.

FAU_STG.4.1 The TSF shall take [notify authorized administrator] if the audit data storage exceeds [DBMS warning notification threshold (default: 90%, configurable range by authorized administrator: 60% ~ 99%)].

6.1.1.8 FAU_STG.5 Prevention of audit data loss

Hierarchical to FAU_STG.4 Action in case of possible audit data loss
Dependencies FAU_STG.2 Protected audit data storage

FAU_STG.5.1 The TSF shall *overwrite the oldest audit data*, [send an email to the authorized administrator] if the audit data storage is full.

6.1.2 Cryptographic support (FCS)

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

Hierarchical to to No other components.
Dependencies [FCS_CKM.2 Cryptographic key distribution or
FCS_CKM.5 Cryptographic key derivation or
FCS_COP.1 Cryptographic operation]
FCS_CKM.3 Cryptographic key access.
[FCS_RBG.1 Random bit generation or
FCS_RNG.1 Random number generation]
FCS_CKM.6 Cryptographic key destruction time and incident.

FCS_CKM.1.1 The TSF shall generate **Data Encryption Keys (DEK)** according to the specified cryptographic key generation algorithm [HASH_DRBG(SHA2-256)] that complies with [TTAK.KO-12.0331].

[Table 11] Data Encryption Key (DEK) Cryptographic Key Generation Algorithm and Key Size List

Standard List	Cryptographic Key Generation Algorithm	Cryptographic Key Length
TTAK.KO-12.0331	HASH_DRBG(SHA256)	128 bits
		192 bits
		256 bits

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

Hierarchical to No other components.
Dependencies [[FCS_CKM.2 Cryptographic key distribution or
FCS_CKM.5 Cryptographic key derivation or
FCS_COP.1 Cryptographic operation].
[FCS_RBG.1 Random bit generation or

FCS_RNG.1 Random number generation].
FCS_CKM.6 Cryptographic key destruction time and incident.

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [Cryptographic key generation algorithm in table below] and specified cryptographic key sizes [Cryptographic key length in table below] that meet the following: [Standard list in table below].

[Table 12] Cryptographic Key Generation Algorithm and Key Size List

Cryptographic Key Classification	Standard List	Cryptographic Key Generation Algorithm	Cryptographic Key Length
TSF data encryption key	TTAK.KO-12.0331	HASH_DRBG(SHA256)	256 bits
Public key/private key	KS X ISO/IEC 18033-2	RSAES(SHA256)	2048 bits
Signature key/verification key	KS X ISO/IEC 14888-2	RSA-PSS(SHA256)	2048 bits
Session key (symmetric key)	TTAK.KO-12.0331	HASH_DRBG(SHA256)	256 bits

6.1.2.3 FCS_CKM.2 Cryptographic key distribution

Hierarchical to No other components.
Dependencies [FDP_ITC.1 Import of user data without security attributes or
FDP_ITC.2 Import of user data with security attributes or
FCS_CKM.1 Cryptographic key generation or
FCS_CKM.5 Cryptographic key derivation]

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method [transfer user data encryption DEK encrypted with ARIA-256(CBC) to API by RSA-OAEP encryption] that meets the following: [None]

6.1.2.4 FCS_CKM.5 Cryptographic key derivation

Hierarchical to No other components.
Dependencies [FCS_CKM.2 Cryptographic key distribution or

FCS_COP.1 Cryptographic operation]
FCS_CKM.6 Cryptographic key destruction time and incident.

FCS_CKM.5.1 The TSF shall derive cryptographic keys [Key Encryption Key (KEK)] from [administrator-entered password, Salt generated via HASH_DRBG, iteration count, derived key length] in accordance with a specified key derivation algorithm [PBKDF2(SHA-256)] and specified cryptographic key sizes [256 bits] that meet the following: [TTAK.KO-12.0334]

[Table 13] Key Encryption Key (KEK) Key Derivation Algorithm and Key Size List

Standard List	Key Derivation Algorithm	Cryptographic Key Length
TTAK.KO-12.0334	PBKDF2(SHA-256)	256 bits

6.1.2.5 FCS_CKM.6 Timing and event of cryptographic key destruction

Hierarchical to No other components.

Dependencies [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation, or
FCS_CKM.5 Cryptographic key derivation]

FCS_CKM.6.1 The TSF shall destroy [KEK, DEK] when *no longer needed.*

FCS_CKM.6.2 The TSF shall destroy cryptographic keys and keying material specified by FCS_CKM.6.1 in accordance with a specified cryptographic key destruction method [Overwrite cryptographic key memory with "0x00" value 3 times] that meets the following: [None]

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

Hierarchical to No other components.

Dependencies [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation, or
FCS_CKM.5 Cryptographic key derivation]
FCS_CKM.6 Timing and event of cryptographic key destruction

FCS_COP.1.1 The TSF shall perform [user data encryption/decryption] in accordance with a specified cryptographic algorithm [ARIA] and cryptographic key sizes [128, 192,

256 bits] that meet the following: [KS X ISO/IEC 1213-1].

[Table 14] User Data Cryptographic Operation Standards and Algorithms

Standard List	Cryptographic Algorithm	Cryptographic Key Length	Operation Mode	Cryptographic Operation List
KS X 1213-1	ARIA	128, 192, 256	CBC	사용자 데이터 암/복호화

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

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation, or
FCS_CKM.5 Cryptographic key derivation]
FCS_CKM.6 Timing and event of cryptographic key destruction

FCS_COP.1.1: The TSF shall perform [audit records, settings, RSAES private key, RSA-PSS verification key, user data encryption DEK), DEK encryption/decryption] in accordance with a specified cryptographic algorithm [ARIA] and cryptographic key sizes [256 bits] that meet the following: [KS X ISO/IEC 1213-1].

[Table 15] TSF Data Cryptographic Operation Standards and Algorithms

Standard List	Cryptographic Algorithm	Cryptographic Key Length	Operation Mode	Cryptographic Operation List
KS X 1213-1	ARIA	256	CBC	TSF data (audit records, settings, RSAES private key, RSA-PSS verification key, user data encryption DEK), DEK encryption/decryption

6.1.2.8 FCS_COP.1(3) Cryptographic operation (Hash)

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 (TSF data encryption), or
FCS_CKM.5 Cryptographic key derivation]
FCS_CKM.6 Timing and event of cryptographic key destruction

FCS_COP.1.1 The TSF shall perform [integrity verification, administrator password storage] in accordance with a specified cryptographic algorithm [SHA256] and cryptographic key sizes [None] that meet the following: [KS X ISO/IEC 10118-3].

6.1.2.9 FCS_COP.1(4) Cryptographic operation (Digital signature generation)

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 (TSF data encryption), or
FCS_CKM.5 Cryptographic key derivation]
FCS_CKM.6 Timing and event of cryptographic key destruction

FCS_COP.1.1 The TSF shall perform [RSA-PSS signature generation] in accordance with a specified cryptographic algorithm [RSA-PSS] and cryptographic key sizes [2048 bits] that meet the following: [KS X ISO/IEC 14888-2].

6.1.2.10 FCS_COP.1(5) Cryptographic operation (Digital signature verification)

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 (TSF data encryption), or
FCS_CKM.5 Cryptographic key derivation]
FCS_CKM.6 Timing and event of cryptographic key destruction

FCS_COP.1.1 The TSF shall perform [RSA-PSS signature verification] in accordance with a specified cryptographic algorithm [RSA-PSS] and cryptographic key sizes [2048 bits] that meet the following: [KS X ISO/IEC 14888-2].

6.1.2.11 FCS_COP.1(6) Cryptographic operation (Public key 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 (TSF data encryption), or
FCS_CKM.5 Cryptographic key derivation]
FCS_CKM.6 Timing and event of cryptographic key destruction

FCS_COP.1.1 The TSF shall perform [RSA-OAEP encryption] in accordance with a specified cryptographic algorithm [RSAES] and cryptographic key sizes [2048 bits] that meet the following: [KS X ISO/IEC 18033-2].

6.1.2.12 FCS_COP.1(7) Cryptographic operation (Public key decryption)

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 (TSF data encryption), or
FCS_CKM.5 Cryptographic key derivation]
FCS_CKM.6 Timing and event of cryptographic key destruction

FCS_COP.1.1 The TSF shall perform [RSA-OAEP decryption] in accordance with a specified cryptographic algorithm [RSAES] and cryptographic key sizes [2048 bits] that meet the following: [KS X ISO/IEC 18033-2].

6.1.2.13 FCS_RBG.1 Random bit generation (RBG)

Hierarchical to No other components.

Dependencies [FCS_RBG.2 Random bit generation (external seeding), or
FCS_RBG.3 Random bit generation (internal seeding – single source)]
FPT_FLS.1 Failure with preservation of secure state
FPT_TST.1 TSF self-testing

FCS_RBG.1.1 The TSF shall perform deterministic random bit generation services using [Hash_DRBG(SHA256)] in accordance with [TTAK.KO-12.0331] after initialization.

FCS_RBG.1.2 The TSF shall use a *TSF entropy source* [urandom result value] for initialization and reseeding.

FCS_RBG.1.3 The TSF shall update the DRBG state by *de-instantiating and re-instantiating using* a *TSF entropy source* [urandom result value] in accordance with [TTAK.KO-12.0331] in the following situations:

o When the following situations are *required*:

- When generating salt for password derivation.
- When generating salt and public key/private key parameters for public key/private key generation.
- When generating salt and signature key/verification key parameters for signature key/verification key generation.

- When generating symmetric keys and IVs.
- When generating nonce for mutual authentication.

6.1.2.14 FCS_RBG.3 Random bit generation (Internal seeding – Single source)

Hierarchical to No other components
Dependencies FCS_RBG.1 Random bit generation (RBG)

FCS_RBG.3.1 The TSF shall be able to seed the DRBG using a *TSF software-based entropy source* [urandom result value] with [40] bits of min-entropy.

6.1.3 User data protection (FDP)

6.1.3.1 FDP_UDE.1 User data encryption

Hierarchical to No other components
Dependencies FCS_COP.1 Cryptographic operation

FDP_UDE.1.1 TSF shall provide TOE users with the ability to encrypt/decrypt user data according to [column-specific encryption/decryption method, [None]] specified

6.1.3.2 FDP_RIP.1 Subset residual information protection

Hierarchical to No other components
Dependencies No dependencies.

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the *allocation of the resource to, deallocation of the resource from* the following objects: [user data]

6.1.3.3 Identification and authentication (FIA)

6.1.3.4 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 [account lockout for the administrator-configured deactivation time (default: 5 minutes, 5~10 minutes)].

6.1.3.5 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 [Alpha DBGuard API, Alpha DBGuard Manager] using the [self-implemented authentication protocol] that meets the following [None].

6.1.3.6 FIA_SOS.1 Verification of secrets

Hierarchical to No other components

Dependencies FIA_UAU.1 Timing of authentication

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet [the acceptance criteria defined in [Table 11] below].

[Table 11] Secret Information Acceptance Criteria

Allowed Characters	Uppercase letters (26)	A - Z
	Lowercase letters (26)	a – z
	Numbers (10)	0 – 9
	Special characters	~ ! @ # $ % ^ & * ( ) _ - + = [ ] { } | ₩ : < > , . ? /
Password Combination Rules	Must include at least one uppercase letter, one lowercase letter, one number, and one special character (4 combinations)	
	[Administrator password] and [User password] - 9 to 20 characters - Includes at least one uppercase/lowercase letter, number, and special character each - Allows a maximum of 2 identical characters and numbers - Allows a maximum of 2 consecutive characters (including forward and reverse) and numbers in keyboard layout - Prohibits setting the same value as the ID	

	- Prohibits using the same password as the previous password

6.1.3.7 FIA_UAU.2 User authentication before any action

Hierarchical to FIA_UAU.1 Authentication.
Dependencies FIA_UID.1 Identification.

FIA_UAU.2.1 The TSF shall successfully authenticate the user before allowing all other TSF-mediated actions on behalf of the user.

6.1.3.8 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 [administrator password authentication, SMTP password authentication, server session].

6.1.3.9 FIA_UAU.7 Protected authentication feedback

Hierarchical to No other components.
Dependencies FIA_UAU.1 Timing of authentication

FIA_UAU.7.1 The TSF shall provide only ['*' character, "Invalid Login Info" error message] to the user while the authentication is in progress.

6.1.3.10 FIA_UID.2 User identification prior to all actions

Hierarchical to FIA_UID.1
Dependencies No dependencies.

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

6.1.4 Security Management (FMT)

6.1.4.1 FMT_MOF.1 Management of security functions behavior

Hierarchical to No other components.

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

FMT_MOF.1.1 The TSF shall restrict the ability to conduct management actions of the functions [[Table 12] Security management function list below] to [authorized administrators].

[Table 12] Security Management Function List

Security Management Function	Management Action
Initial setup and modification of administrator password	Modify
Setting and modification of audit record storage threshold	View, Modify
Modification of authentication failure count and administrator account deactivation time	View, Modify
Viewing of audit records	View
Setting and modification of cryptographic policies	Create, View, Modify, Delete
Setting and modification of cryptographic keys	Create, View, Modify

6.1.4.2 FMT_MTD.1 Management of TSF data

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

FMT_MTD.1.1 The TSF shall restrict the ability to manage [[Table 13] TSF data list below] to [authorized administrators].

[Table 13] TSF Data List

TSF Data List	Management
Administrator account information	View
Administrator access IP	View, Modify
SMTP access account information	View, Modify
Setting of audit information storage threshold	View, Modify
Encryption policy	Create, View, Modify, Delete
Cryptographic key (User data encryption)	Create, View, Modify
Audit information	View

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

Hierarchical to No other components.
Dependencies FMT_SMF.1 Specification of Management Functions

FMT_SMR.1 Security roles

FMT_PWD.1.1 The TSF shall restrict the ability to manage the password of [administrator password change function] to [the authorized administrator].
1. [Password combination rules according to secret information acceptance criteria in [Table 11]].
2. [None]

FMT_PWD.1.2 The TSF shall restrict the ability to manage the ID of [None] to [authorized administrators].
1. [None]
2. [None]

FMT_PWD.1.3 The TSF shall provide the capability for *authorized administrators to change their password upon first access*.

6.1.4.4 FMT_SMF.1 Specification of Management Functions

Hierarchical to No other components.

Dependencies No dependencies.

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: [list of management functions provided by the TSF below]
- TSF Function Management: Management function list described in FMT_MOF.1.
- TSF Data Management: Management function list described in FMT_MTD.1.
- ID and Password Management: Management function list described in FMT_PWD.1.

6.1.4.5 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 [the following administrator].
[Initial setup and modification of administrator password
Setting and modification of audit record storage threshold
Modification of authentication failure count and administrator account deactivation time
Viewing of audit records
Setting and modification of cryptographic policies
Setting and modification of cryptographic keys]

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

6.1.5 Protection of the TSF (FPT)

6.1.5.1 FPT_ITT.1 Basic internal TSF data transfer protection

Hierarchical to No other components.
Dependencies No dependencies.

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

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

Hierarchical to No other components.
Dependencies No dependencies.

FPT_PST.1.1 The TSF shall protect [the TSF data in Table 22] stored in containers controlled by the TSF from the unauthorized *disclosure*

[Table 22] TSF Data Storage Cryptographic Algorithm

Classification	TSF Data	Encryption Algorithm	Remarks
TOE Server	Password used by TOE for user identification and authentication	SHA256	During hashing, add 128 bits generated using Hash_DRBG(SHA256) to the password, apply an iteration count of 1000, and store in the DBMS interworking with the TOE (protected from unauthorized user access using DBMS identification and authentication functions))
	Original hash value for integrity verification target	ARIA-256(CBC)	Encrypted with TSF data encryption DEK and stored in file system (/asm/integrity/)
	DBMS access information (ID/PW)	ARIA-256(CBC)	Encrypted with TSF data encryption DEK and stored in file system

			(/asm/conf/application.properties)
	SMTP settings (ID, PW)	ARIA-256(CBC)	Encrypted with TSF data encryption DEK and stored in DBMS (protected from unauthorized user access using DBMS identification and authentication functions)
	KEK	-	Hash_DRBG(SHA256) used to generate 128-bit salt stored in file system (/asm/keystore/salt) KEK is not stored, but loaded into memory and destroyed from memory when the TOE server terminates
	User data encryption DEK	ARIA-256(CBC)	Encrypted with KEK, base64 encoded, and stored in DBMS (protected from unauthorized user access using DBMS identification and authentication functions)
	TSF data encryption DEK	ARIA-256(CBC)	Encrypted with KEK and stored in file system (/asm/keystore/dekServer)
	RSAES private key	ARIA-256(CBC)	Encrypted with KEK and stored in file system (/asm/keystore/server_key_enc.pri)
	RSA-PSS verification key	ARIA-256(CBC)	Encrypted with KEK and stored in file system (/asm/keystore/server_key_sig.pri)
TOE API	Original hash value for integrity verification target	ARIA-256(CBC)	Encrypted with DEK and stored in file system (alpha/smart/sapi_root/hashList)
	KEK	-	Store administrator-entered password, salt generated using Hash_DRBG(SHA256), IV by self-encoding (/api/sapi_root/keystore/master_key) KEK is not stored, but loaded into memory and destroyed from memory when the API terminates
	TSF data encryption DEK	ARIA-256(CBC)	Encrypted with KEK and stored in file system (/api/sapi_root/keystore/conf.key)
	RSAES private key	ARIA-256(CBC)	Encrypted with KEK and stored in file system (/api/sapi_root/keystore/client_key1_enc.p

			ri)
	RSA-PSS verification key	ARIA-256(CBC)	Encrypted with KEK and stored in file system (/api/sapi_root/keystore/client_key1_sig.pri)

6.1.5.3 FPT_TST.1 TSF testing

Hierarchical to No other components.
Dependencies No dependencies.

FPT_TST.1.1 The TSF shall run a suite of the following self-tests during *initial start-up, periodically during normal operation, at the request of the authorized administrator* to demonstrate the correct operation of the *TSF*: [[Table 14] Test Content].

[Table 14] TOE Self-Test Targets

Self-Test Item	Test Content
Execution Process Manager	Performs self-test at startup and generates audit logs. Records audit log and stops process if self-test fails. Monitors if processes are running normally every hour during operation.

FPT_TST.1.2 The TSF shall provide authorized administrators with the capability to verify the integrity of *TSF data*

FPT_TST.1.3 The TSF shall provide authorized administrators with the capability to verify the integrity of *TSF*.

6.1.5.4 FPT_FLS.1 Failure with preservation of secure state

Hierarchical to No other components.
Dependencies No dependencies.

FPT_FLS.1.1 The TSF shall maintain a secure state upon occurrence of the following types of failures:
[Transition to severe error and application operation stop upon random number health test failure.
Failure situations occur during the following function operations:
- When generating Salt for password derivation.
- When generating salt and public key/private key parameters for public

key/private key generation.

- When generating symmetric keys and IVs.
- When generating nonce for mutual authentication.

]

6.1.6 TOE Access (FTA)

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

Hierarchical to FTA_MCS.1 Basic limitation on multiple concurrent sessions

Dependencies FIA_UID.1 Timing of identification

FTA_MCS.2.1 The TSF shall restrict the maximum number of concurrent sessions belong to the same user according to the rules [maximum number of concurrent sessions for users with the same privileges and for the same user is limited to 1, [None]].

FTA_MCS.2.2 The TSF shall enforce, by default, a limit of [1] sessions per user.

6.1.6.2 FTA_SSL.3 TSF-initiated termination

Hierarchical to No other components.

Dependencies FMT_SMR.1 Security roles

FTA_SSL.3.1 The TSF shall terminate interactive sessions after [**administrator** inactivity for 10 minutes].

6.1.6.3 FTA_TSE.1 TOE TOE session establishment

Hierarchical to No other components.

Dependencies No dependencies.

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

6.1.7 Secure Path/Channel (FTP)

6.1.7.1 FTP_ITC.1 Inter-TSF trusted channel

Hierarchical to No other components.

Dependencies No dependencies.

FTP_ITC.1.1 The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels, provides assured identification of the channel endpoint, and protects channel data from modification or disclosure.

FTP_ITC.1.2 The TSF shall permit *TSF* to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for [E-Mail transmission function using SMTP].

6.2 Security assurance requirements

The assurance requirements of this Security Target are composed of assurance components from CC Part 3, and the Evaluation Assurance Level is EAL1+ (ATE_FUN.1). The following [Table 15] summarizes the assurance components

[Table 15] Security Assurance Requirements

Assurance Class	Assurance Components	
Security Target	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	Direct rationale 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

6.2.1 Security Target evaluation

6.2.1.1 ASE_INT.1 ST introduction

Dependencies No dependencies.

Developer action elements

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

Content and presentation elements

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

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

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

ASE_INT.1.4C The TOE overview shall summaries 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 For a multi-assurance ST, the TOE overview shall describe the TSF organization in terms of the sub-TSFs defined in the PP-Configuration the ST claims conformance to.

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

ASE_INT.1.9C 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.

6.2.1.2 ASE_CCL.1 Conformance claims

Dependencies ASE_INT.1 ST introduction
ASE_ECD.1 Extended components definition
ASE_REQ.1 Direct rationale 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 identify the edition 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 a PP-Configuration, or all PPs and security requirement packages to which the ST claims conformance 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 shall describe any conformance of the ST to a PP as PP-Conformant.

ASE_CCL.1.8C The conformance claim rationale shall demonstrate that the TOE type is consistent with the TOE type in the PP-Configuration or PPs for which conformance is being claimed.

ASE_CCL.1.9C 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 PP-Configuration, PPs and any functional packages for which conformance is being claimed.

ASE_CCL.1.10C The conformance claim rationale shall demonstrate that the statement of security objectives is consistent with the statement of security objectives in the PP-Configuration, PPs, and any functional package for which conformance is being claimed.

ASE_CCL.1.11C The conformance claim rationale shall demonstrate that the statement of security requirements is consistent with the statement of security requirements in the PP-Configuration, PPs, and any functional packages for which conformance is being claimed.

ASE_CCL.1.12C The conformance claim for PP(s) or a PP-Configuration shall be exact, strict, or demonstrable or a list of conformance types.

ASE_CCL.1.13C If the conformance claim identifies a set of Evaluation methods and Evaluation activities derived from CEM work units that shall be used to evaluate the TOE then this set shall include all those that are included in any package, PP, or PP-Module in a PP-Configuration to which the ST claims conformance, and no others.

Evaluator action elements

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

6.2.1.3 ASE_OBJ.1 Security objectives for the operational environment

Dependencies ASE_SPD.1 Security problem definition.

Developer action elements

ASE_OBJ.1.1D The developer shall provide a statement of security objectives for the operational environment.

ASE_OBJ.1.2D The developer shall provide a security objectives rationale for the operational environment.

Content and presentation elements

ASE_OBJ.1.1C The statement of security objectives shall describe the security objectives for the operational environment.

ASE_OBJ.1.2C The security objectives rationale shall trace each security objective for the operational environment back to threats countered by that security objective, OSPs enforced by that security objective, and assumptions upheld by that security objective.

ASE_OBJ.1.3C The security objectives rationale shall demonstrate that the security objectives for the operational environment uphold all assumptions.

Evaluator action elements

ASE_OBJ.1.1E The evaluator shall confirm that the information provided meets all requirements

for content and presentation of evidence.

6.2.1.4 ASE_ECD.1 Extended components definition

Dependencies No dependencies.

Developer action elements

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

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

Content and presentation elements

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

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

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

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

ASE_ECD.1.5C The extended components shall consist of measurable and objective elements such that conformance or nonconformance to these elements may 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.

6.2.1.5 ASE_REQ.1 Direct rationale security requirements

Dependencies ASE_ECD.1 Extended components definition
ASE_SPD.1 Security problem definition
ASE_OBJ.1 Security objectives for the operational environment

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 For a single-assurance ST, the statement of security requirements shall define the global set of SARs that apply to the entire TOE. The sets of SARs shall be consistent with the PPs or PP-Configuration to which the ST claims conformance.

ASE_REQ.1.3C For a multi-assurance ST, the statement of security requirements shall define the global set of SARs that apply to the entire TOE and the sets of SARs that apply to each sub-TSF. The sets of SARs shall be consistent with the multi-assurance PP-Configuration to which the ST claims conformance.

ASE_REQ.1.4C 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.1.5C The statement of security requirements shall identify all operations on the security requirements.

ASE_REQ.1.6C All operations shall be performed correctly.

ASE_REQ.1.7C Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.

ASE_REQ.1.8C The security requirements rationale shall demonstrate that the SFRs (in conjunction with the security objectives for the environment) counter all threats for the TOE.

ASE_REQ.1.9C The security requirements rationale shall demonstrate that the SFRs (in conjunction with the security objectives for the environment) enforce all OSPs.

ASE_REQ.1.10C The security requirements rationale shall explain why the SARs were chosen.

ASE_REQ.1.11C The statement of security requirements shall be internally consistent.

ASE_REQ.1.12C If the ST defines sets of SARs that expand the sets of SARs of the PPs or PP-Configuration it claims conformance to, the security requirements rationale shall include an assurance rationale that justifies the consistency of the extension and provides a rationale for the disposition of any Evaluation methods and Evaluation activities identified in the conformance statement that are affected by the extension of the sets of SARs.

Evaluator action

elements

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

6.2.1.6 ASE_TSS.1 TOE summary specification

Dependencies ASE_INT.1 ST introduction
ASE_REQ.1 Direct rationale 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.

6.2.2 Development

6.2.2.1 ADV_FSP.1 Basic functional specification

Dependencies No dependencies.

Developer action elements

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

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

Content and presentation elements	
ADV_FSP.1.1C	The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.
ADV_FSP.1.2C	The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.
ADV_FSP.1.3C	The functional specification shall provide rationale for the implicit categorization of interfaces as SFR-non-interfering.
ADV_FSP.1.4C	The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.
Evaluator action elements	
ADV_FSP.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_FSP.1.2E	The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFR.

6.2.3 Guidance documents

6.2.3.1 AGD_OPE.1 Operational user guidance

Dependencies	ADV_FSP.1 Basic functional specification
Developer action elements	
AGD_OPE.1.1D	The developer shall provide operational user guidance.
Content and presentation elements	
AGD_OPE.1.1C	The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that shall 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 controls 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.

6.2.3.2 AGD_PRE.1 Preparative procedures

Dependencies No dependencies.

Developer action elements

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

Content and presentation elements

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

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

Evaluator action elements

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

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

6.2.4 Life-cycle support

6.2.4.1 ALC_CMC.1 Labelling of the TOE

Dependencies ALC_CMS.1 TOE CM coverage

Developer action elements

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

6.2.4.2 ALC_CMS.1 TOE CM coverage

Dependencies No dependencies.

Developer action elements

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

Content and presentation elements

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

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

Evaluator action elements

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

6.2.5 Tests

6.2.5.1 ATE_FUN.1 Functional testing

Dependencies ATE_COV.1 Evidence of coverage

Developer action elements

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

ATE_FUN.1.2D The developer shall provide test documentation.

Content and presentation elements

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

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

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

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

Evaluator action elements

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

6.2.5.2 ATE_IND.1 Independent testing - conformance

Dependencies	ADV_FSP.1 Basic functional specification AGD_OPE.1 Operational user guidance AGD_PRE.1 Preparative procedures
Developer action elements	
ATE_IND.1.1D	The developer shall provide the TOE 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.

6.2.6 Vulnerability assessment

6.2.6.1 AVA_VAN.1 Vulnerability survey

Dependencies	ADV_FSP.1 Basic functional specification AGD_OPE.1 Operational user guidance AGD_PRE.1 Preparative procedures
Developer action elements	
AVA_VAN.1.1D	The developer shall provide the TOE 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.

6.3 Security Requirements Rationale

The rationale for the security requirements demonstrates that the described security requirements are appropriate for satisfying the security objectives and, as a result, for addressing the security problems.

6.3.1 Security Functional Requirements Rationale

The rationale for the security functional requirements demonstrates the following:

- Each threat and organizational security policy is addressed by at least one SFR.
- Each SFR is addressed by at least one threat or OSP for the TOE.

SFR	T. SESSION_HIJACK	T. RETRY_AUTH_ATTE MPT	T. IMPE0RSONATION	T. REPLAY	T. WEAK_PASSWORD	T. UNAUTHORIZED_INFO_LEA	T. STORED_DATA_LEAKAGE	T. TRANSMISSION_DATA_DAMAGE	T. WEAK_CRYPTO_PROTOCOLS	T. TSF_COMPROMISE	P. AUDIT	P. SECURE_OPERATION	P. CRYPTO_STRENGTH
FAU_ARP.1										X			
FAU_GEN.1											X		
FAU_SAA.1										X			
FAU_SAR.1											X		
FAU_SAR.3											X		
FAU_STG.1											X		
FAU_STG.4											X		
FAU_STG.5											X		
FCS_CKM.1(1)						X							X
FCS_CKM.1(2)							X	X	X				X
FCS_CKM.2						X	X	X	X				X
FCS_CKM.5						X	X	X	X				X
FCS_CKM.6						X	X	X	X				X
FCS_COP.1(1)						X							X
FCS_COP.1(2)							X	X	X				X
FCS_COP.1(3)							X	X	X				X
FCS_COP.1(4)							X	X	X				X
FCS_COP.1(5)							X	X	X				X
FCS_COP.1(6)							X	X	X				X
FCS_COP.1(7)							X	X	X				X
FCS_RBG.1						X	X	X	X				X
FCS_RBG.3						X	X	X	X				X

FDP_UDE.1						X							
FDP_RIP.1						X							
FIA_AFL.1		X	X							X			
FIA_IMA.1(Extended)			X										
FIA_SOS.1					X								
FIA_UAU.2			X							X			
FIA_UAU.4			X	X						X			
FIA_UAU.7			X		X					X			
FIA_UID.2			X							X			
FMT_MOF.1										X		X	
FMT_MTD.1										X		X	
FMT_PWD.1(Extended)					X					X		X	
FMT_SMF.1										X		X	
FMT_SMR.1										X		X	
FPT_FLS.1						X	X	X	X				X
FPT_ITT.1								X					
FPT_PST.1(Extended)							X						
FPT_TST.1						X	X	X	X	X			X
FTA_MCS.2	X												
FTA_SSL.3	X												
FTA_TSE.1	X												
FTP_ITC.1								X					

T.SESSION_HIJACK FTA_MCS.2, FTA_SSL.3, FTA_TSE.1

FTA_MCS.2 responds to T.SESSION_HIJACK by restricting concurrent access to the TOE with the same user account or same privileges.

FTA_SSL.3 respond to T.SESSION_HIJACK by ensuring session locking or session termination for interactive sessions after a period of inactivity by authorized users.

FTA_TSE.1 respond to T.SESSION_HIJACK by ensuring that it determines whether to establish an authorized user access session based on IP, etc.

T.RETRY_AUTH_ATTEMPT FIA_AFL.1

FIA_AFL.1 responds to T.RETRY_AUTH_ATTEMPT by defining the number of failed authentication attempts by authorized users and ensuring the ability to take responsive action when the defined number is reached.

T.IMPERSONATION FIA_AFL.1, FIA_IMA.1, FIA_UAU.2, FIA_UAU.4, FIA_UAU.7, FIA_UID.2

FIA_AFL.1 responds to T.IMPERSONATION by defining the number of failed authentication

attempts by authorized users and ensuring the ability to take responsive action when the defined number is reached.
FIA_IMA.1 responds to T.IMPERSONATION by ensuring that mutual authentication is conducted between TOE components.
FIA_UAU.2, FIA_UAU.4 respond to T.IMPERSONATION by ensuring that users attempting to access the TOE are successfully authenticated.
FIA_UAU.7 responds to T.IMPERSONATION by ensuring that only masked values will be output or no display to users during authentication and not providing feedback on the reason for failure in case of authentication failure.
FIA_UID.2 responds to T.IMPERSONATION by ensuring that users attempting to access the TOE are successfully identified

T.REPLAY FIA_UAU.4
FIA_UAU.4 responds to T.REPLAY by ensuring the ability to prevent reuse of authentication data.

T.WEAK_PASSWORD FIA_UAU.7, FIA_SOS.1, FMT_PWD.1
FIA_UAU.7 responds to T.WEAK_PASSWORD by ensuring that only masked values will be output or no display to users during authentication.
FIA_SOS.1 responds to T.WEAK_PASSWORD by verifying that password complexity rules are satisfied.
FMT_PWD.1 responds to T.WEAK_PASSWORD by ensuring the ability to force a change of the default password when the authorized administrator first connects

T.UNAUTHORIZED_INFO_LEAK FCS_CKM.1(1), FCS_CKM.2, FCS_CKM.5, FCS_CKM.6, FCS_COP.1(1), FCS_RBG.1, FCS_RBG.3, FDP_UDE.1, FDP_RIP.1, FPT_FLS.1, FPT_TST.1
FCS_CKM.1(1), FCS_CKM.2, FCS_CKM.5, FCS_RBG.1, FCS_RBG.3, FPT_FLS.1 and FPT_TST.1 respond to T.UNAUTHORIZED_INFO_LEAK by ensuring that a cryptographic key is created and distributed according to a secure cryptographic algorithm and key length when encrypting and decrypting important user information stored in the database.
FCS_CKM.6 responds to T.UNAUTHORIZED_INFO_LEAK by ensuring that cryptographic keys are destroyed according to the specified cryptographic key destruction method after encrypting and decrypting important user information stored in the database.
FCS_COP.1(1) responds to T.UNAUTHORIZED_INFO_LEAK by ensuring that cryptographic operations are performed according to the specified secure algorithm and cryptographic key length when encrypting and decrypting important user information stored in the database.
FDP_RIP.1 responds to T.UNAUTHORIZED_INFO_LEAK by ensuring that all original user data is deleted after the encrypting and decrypting important user information stored in the database.
FDP_UDE.1 responds to T.UNAUTHORIZED_INFO_LEAK by ensuring that encryption and decryption

occur when an authorized user stores important information in the database

T. STORED_DATA_LEAKAGE	FCS_CKM.1(2), FCS_CKM.2, FCS_CKM.5, FCS_CKM.6, FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_COP.1(5), FCS_COP.1(6), FCS_COP.1(7), FCS_RBG.1, FCS_RBG.3, FPT_FLS.1, FPT_PST.1, FPT_TST.1

FCS_CKM.1(1), FCS_CKM.2, FCS_CKM.5, FCS_RBG.1, FCS_RBG.2, FCS_RBG.3, FPT_FLS.1 and FPT_TST.1 respond to T.UNAUTHORIZED_INFO_LEAK by ensuring that a cryptographic key is created and distributed according to a secure cryptographic algorithm and key length when encrypting and decrypting important user information stored in the database.

FCS_CKM.6 responds to T.UNAUTHORIZED_INFO_LEAK by ensuring that cryptographic keys are destroyed according to the specified cryptographic key destruction method after encrypting and decrypting important user information stored in the database.

FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_COP.1(5), FCS_COP.1(6) and FCS_COP.1(7) responds to T.UNAUTHORIZED_INFO_LEAK by ensuring that cryptographic operations are performed according to the specified secure algorithm and cryptographic key length when encrypting and decrypting important user information stored in the database.

FPT_PST.1 responds to T.STORED_DATA_LEAKAGE by ensuring that the stored TSF data is protected from being leaked by means of encryption, access control, etc.

T.TRANSMISSION_DATA_DAMAGE	FCS_CKM.1(2), FCS_CKM.2, FCS_CKM.5, FCS_CKM.6, FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_COP.1(5), FCS_COP.1(6), FCS_COP.1(7), FCS_RBG.1, FCS_RBG.3, FPT_FLS.1, FPT_ITT.1, FPT_TST.1 FTP_ITC.1

FCS_CKM.1(2), FCS_CKM.2, FCS_CKM.5, FCS_RBG.1, FCS_RBG.2, FCS_RBG.3, FPT_FLS.1, and FPT_TST.1 respond to T.TRANSMISSION_DATA_DAMAGE by ensuring that a cryptographic key is created and distributed according to a secure cryptographic algorithm and key length during cryptographic communication.

FCS_CKM.6 responds to T.TRANSMISSION_DATA_DAMAGE by ensuring that cryptographic keys and their related information are destroyed according to the specified cryptographic key destruction method at the end of cryptographic communication.

FCS_COP.1 responds to T.TRANSMISSION_DATA_DAMAGE by ensuring that cryptographic operations are performed according to the specified secure algorithm and specified cryptographic key length during cryptographic communication.

FPT_ITT.1 responds to T.TRANSMISSION_DATA_DAMAGE by ensuring the confidentiality and integrity of transmission data between TOE components.

FTP_ITC.1 responds to T.TRANSMISSION_DATA_DAMAGE by ensuring the confidentiality and integrity of transmission data between the TOE and external IT entities.

T.WEAK_CRYPTO_PROTOCOLS FCS_CKM.1(2), FCS_CKM.2, FCS_CKM.5, FCS_CKM.6, FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_COP.1(5), FCS_COP.1(6), FCS_COP.1(7), FCS_RBG.1, FCS_RBG.3, FPT_FLS.1, FPT_TST.1

FCS_CKM.1(2), FCS_CKM.2, FCS_CKM.5, FCS_RBG.1, FCS_RBG.2, FCS_RBG.3, FPT_FLS.1, and FPT_TST.1 respond to T.WEAK_CRYPTO_PROTOCOLS by ensuring that the cryptographic key is created and distributed according to the standard cryptographic algorithm and key length with a security strength of 112 bits or more when encrypting transmission data.

FCS_CKM.6 responds to T.WEAK_CRYPTO_PROTOCOLS by ensuring that the cryptographic key is destroyed according to the specified destruction method.

FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_COP.1(5), FCS_COP.1(6) and FCS_COP.1(7) responds to T.WEAK_CRYPTO_PROTOCOLS by ensuring that cryptographic operations are performed according to the standard cryptographic algorithm and cryptographic key length with a security strength of 112 bits or more when encrypting transmission data.

T.TSF_COMPROMISE FAU_ARP.1, FAU_SAA.1, FIA_AFL.1, FIA_UAU.2, FIA_UAU.4, FIA_UAU.7, FIA_UID.2 FMT_MOF.1, FMT_MTD.1, FMT_PWD.1, FMT_SMF.1, FMT_SMR.1, FPT_TST.1

FAU_ARP.1 responds to T.TSF_COMPROMISE by ensuring the ability to take response actions when detecting security violations such as TOE integrity compromise, etc.

FAU_SAA.1 responds to T.TSF_COMPROMISE by ensuring the ability to review audited events to point out security violations, such as TOE integrity compromise.

FIA_AFL.1, FIA_UAU.2, FIA_UAU.4, FIA_UAU.7 and FIA_UID.2 respond to T.TSF_COMPROMISE by allowing access to the TOE only after successful user identification and authentication, ensuring the blocking of bypass access by threat agents.

FMT_MOF.1, FMT_MTD.1, FMT_PWD.1, FMT_SMF.1, and FMT_SMR.1 respond to T.TSF_COMPROMISE by dividing authorized user roles into administrator and end user when accessing and configuring management functions, and by providing security policies and functions based on those roles to ensure blocking of unauthorized access by threat agents.

FPT_TST.1 responds to T.TSF_COMPROMISE by ensuring the TSF self-testing for accurate operation of the TOE and ensuring that authorized administrators can verify the integrity of TSF data and the TSF itself.

P.AUDIT FAU_GEN.1, FAU_SAR.1, FAU_SAR.3, FAU_STG.1, FAU_STG.4, FAU_STG.5

FAU_GEN.1 satisfies P.AUDIT by ensuring that audit records are generated for auditable events such as the startup/termination of the audit function and the success/failure of the identification and authentication of the administrator.

FAU_SAR.1 satisfies P.AUDIT by providing the authorized administrator with the ability to retrieve audit records and ensuring that the audit records are presented in a manner suitable for the administrator to interpret the information.

FAU_SAR.3 satisfies P.AUDIT by providing a selective audit review function based on logical relationship criteria for audit data.
FAU_STG.1 satisfies P.AUDIT by providing the ability to store audit data in local storage or transmit it to an external IT entity for storage in real time using a trusted channel for the TOE server.
FAU_STG.4 satisfies P.AUDIT by ensuring that appropriate response actions are taken if the audit trail on the TOE server exceeds the storage limit.
FAU_STG.5 satisfies P.AUDIT by ensuring the ability to take appropriate response actions when the audit trail of the TOE server is full.

P.SECURE_OPERATION	FMT_MOF.1, FMT_MTD.1, FMT_PWD.1, FMT_SMF.1, FMT_SMR.1

FMT_MOF.1 satisfies P.SECURE_OPERATION by ensuring that only authorized users have the ability to manage security functions.
FMT_MTD.1 satisfies P.SECURE_OPERATION by ensuring that only authorized users have the ability to manage the TSF data.
FMT_PWD.1 satisfies P.SECURE_OPERATION by ensuring that only authorized administrators have the ability to manage the combination rules and length of IDs and passwords, and by providing functions such as changing passwords when authorized administrators first access.
FMT_SMF.1 satisfies P.SECURE_OPERATION by requiring management functions such as security functions to be performed by the TSF, the TSF data, etc. to be specified.
FMT_SMR.1 satisfies P.SECURE_OPERATION by ensuring that authorized roles related to security management are specified.

P.CRYPTO_STRENGTH	FCS_CKM.1(1), FCS_CKM.1(2), FCS_CKM.2, FCS_CKM.5, FCS_CKM.6, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_COP.1(5), FCS_COP.1(6), FCS_COP.1(7), FCS_RBG.1, FCS_RBG.3, FPT_FLS.1, FPT_TST.1

FCS_CKM.1(1), FCS_CKM.1(2), FCS_CKM.2, FCS_CKM.5, FCS_CKM.6, FCS_COP.1, FCS_RBG.1, FCS_RBG.2, FCS_RBG.3, FCS_RBG.4, FCS_RBG.5, FPT_FLS.1, and FPT_TST.1 satisfy P.CRYPTO_STRENGTH by ensuring that the cryptographic keys required for standard cryptographic algorithms with a security strength of 112 bits or more are securely generated and distributed during data encryption.
FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_COP.1(5), FCS_COP.1(6) and FCS_COP.1(7) satisfies P.CRYPTO_STRENGTH by ensuring that cryptographic operations are performed according to standard cryptographic algorithms with a security strength of 112 bits or more and the cryptographic key length during data encryption.

6.3.2 Security assurance requirements rationale

The assurance level for this Security Target was selected as EAL1+ in accordance with the "Korean National Protection Profile for Database Encryption V3.1" that the Security Target complies with.

EAL1 can be applied when a certain level of trust in correct operation is required, but security threats are not severe. EAL1 is useful when independent assurance is required to demonstrate that appropriate measures have been taken to protect personal or similar information.

EAL1 requires only a limited ST. That is, EAL1 is sufficient to simply present security functional requirements clearly to the TOE, rather than defining security objectives from threats, organizational security policies (OSP), and assumptions based on security objectives and deriving security functional requirements (SFRs therefrom.

EAL1 does not require evidence of developer-performed testing based on functional specifications. However, this Protection Profile augmented ATE_FUN.1 to enable the developer to self-test and document the results regarding the accurate implementation of the TSF and the occurrence of defects.

6.4 Dependency rationale of security functional requirements

6.4.1 Security functional requirements dependencies

The following table shows the dependencies of the security functional requirements

[표 16] 보안기능요구사항의 종속관계

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.1	FAU_GEN.1, FTP_ITC.1	2, 44
7	FAU_STG.2	FAU_GEN.1	2
8	FAU_STG.4	FAU_STG.2	Rationale(2)
9	FAU_STG.5	FAU_STG.2	Rationale(2)
10	FCS_CKM.1(1)	FCS_CKM.2 or FCS_COP.1(1)	12, 15

		FCS_CKM.6	14
11	FCS_CKM.1(2)	FCS_CKM.2 or FCS_COP.1(2)	12, 16
		FCS_CKM.6	14
12	FCS_CKM.2	FDP_ITC.1 or FCS_CKM.1	10, 11
		FCS_CKM.6	14
13	FCS_CKM.5	FCS_CKM.2 or FCS_COP.1(2)	12, 16
		FCS_CKM.6	14
14	FCS_CKM.6	FDP_ITC.1 or FCS_CKM.1	10, 11
15	FCS_COP.1(1)	FDP_ITC.1 or FCS_CKM.1(1)	10
		FCS_CKM.6	14
16	FCS_COP.1(2)	FDP_ITC.1 or FCS_CKM.1(2)	11
		FCS_CKM.6	14
17	FCS_COP.1(3)	FDP_ITC.1 or FCS_CKM.1(2)	11
		FCS_CKM.6	14
18	FCS_COP.1(4)	FDP_ITC.1 or FCS_CKM.1(2)	11
		FCS_CKM.6	14
19	FCS_COP.1(5)	FDP_ITC.1 or FCS_CKM.1(2)	11
		FCS_CKM.6	14
20	FCS_COP.1(6)	FDP_ITC.1 or FCS_CKM.1(2)	11
		FCS_CKM.6	14
21	FCS_COP.1(7)	FDP_ITC.1 or FCS_CKM.1(2)	11
		FCS_CKM.6	14
22	FCS_RBG.1	FCS_RGB.3	23
		FPT_FLS.1	38
		FPT_TST.1	41
23	FCS_RBG.3	FCS_RBG.1	22
24	FDP_UDE.1	FCS_COP.1(1)	15
25	FDP_RIP.1	-	-
26	FIA_AFL.1	FIA_UAU.2	29
27	FIA_IMA.1	-	-
28	FIA_SOS.1	-	-
29	FIA_UAU.2	FIA_UID.2	32
30	FIA_UAU.4	-	-
31	FIA_UAU.7	FIA_UAU.2	29
32	FIA_UID.2	-	-
33	FMT_MOF.1	FMT_SMF.1	36
		FMT_SMR.1	37

34	FMT_MTD.1	FMT_SMF.1	36
		FMT_SMR.1	37
35	FMT_PWD.1	FMT_SMF.1	36
		FMT_SMR.1	37
36	FMT_SMF.1	-	-
37	FMT_SMR.1	FIA_UID.1	32
38	FTP_FLS.1	-	-
39	FPT_ITT.1	-	-
40	FPT_PST.1	-	-
41	FPT_TST.1	-	-
42	FTA_MCS.2	FIA_UID.1	32
43	FTA_SSL.3	FIA_SMR.1	37
44	FTA_TSE.1	-	-
45	FTP_ITC.1	-	-

Rationale (1): FAU_GEN.1 has a dependency on FPT_STM.1. However, the reliable timestamp provided by the security objective OE.TIME_STAMP for the operational environment in this Security Target is used, thereby satisfying this dependency.

Rationale (2): FAU_STG.4 and FAU_STG.5 have a dependency on FAU_STG.2. However, they are protected from unauthorized deletion or modification by the security objective OE.AUDIT_DATA_PROTECTION for the operational environment in this Security Target, thereby satisfying this dependency.

FIA_AFL.1 and FIA_UAU.7 have a dependency on FIA_UAU.1, but this is satisfied by FIA_UAU.2, which is hierarchical to FIA_UAU.1

FIA_UAU.2, FMT_SMR.1 and FTA_MCS.2 have a dependency on FIA_UID.1, but this is satisfied by FIA_UID.2, which is hierarchical to FIA_UID.1

6.4.2 Security assurance Requirements Dependencies Rationale

No.	Assurance Component	Dependencies	Reference No.
1	ASE_INT.1	-	-
2	ASE_CCL.1	ASE_INT.1 ASE_ECD.1 ASE_REQ.1	1, 5, 6
3	ASE_SPD.1	-	-

4	ASE_OBJ.1	ASE_SPD.1	3
5	ASE_ECD.1	-	-
6	ASE_REQ.1	ASE_ECD.1 ASE_SPD.1 ASE_OBJ.1	5, 3, 4
7	ASE_TSS.1	ASE_INT.1 ASE_REQ.1 ADV_FSP.1	1, 6, 8
8	ADV_FSP.1	-	-
9	AGD_OPE.1	ADV_FSP.1	8
10	AGD_PRE.1	-	-
11	ALC_CMC.1	ALC_CMS.1	12
12	ALC_CMS.1	-	-
13	ATE_FUN.1	ATE_COV.1	rationale(1)
14	ATE_IND.1	ADV_FSP.1 AGD_OPE.1 AGD_PRE.1	8, 9, 10
15	AVA_VAN.1	ADV_FSP.1 AGD_OPE.1 AGD_PRE.1	8, 9, 10

Rationale (1): The augmented assurance requirement ATE_FUN.1 includes ATE_COV.1 as a dependency. ATE_FUN.1 was added to ensure that the developer accurately performs testing for test items and records the results in the test report. It was determined that ATE_COV.1, which presents consistency between test items and TSFI, is not strictly necessary for this Security Target and was therefore not included.

7 TOE Summary Specification

This chapter briefly and clearly describes how the security functions required by the TOE are implemented. The following table lists the security functions specified in the TOE Summary Specification.

[Table 17] Security Functional Components

Functional Class	Security Functional Components	
Security Audit (FAU)	FAU_ARP.1	Security alarms
	FAU_GEN.1	Audit data generation
	FAU_SAA.1	Potential violation analysis
	FAU_SAR.1	Audit review
	FAU_SAR.3	Selectable audit review
	FAU_STG.1	Audit data storage
	FAU_STG.4	Action in case of possible audit data loss
	FAU_STG.5	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.5	Cryptographic key derivation
	FCS_CKM.6	Cryptographic key destruction time and incident
	FCS_COP.1(1)	Cryptographic operation (User data encryption)
	FCS_COP.1(2)	Cryptographic operation (TSF data encryption)
	FCS_COP.1(3)	Cryptographic operation (Hash)
	FCS_COP.1(4)	Cryptographic operation (Digital signature generation)
	FCS_COP.1(5)	Cryptographic operation (Digital signature verification)
	FCS_COP.1(6)	Cryptographic operation (Public key encryption)
	FCS_COP.1(7)	Cryptographic operation (Public key decryption)
	FCS_RBG.1	Random bit generation
	FCS_RBG.3	Random bit generation (Internal seeding – Single source)
User Data Protection (FDP)	FDP_UDE.1(Extended)	User data encryption
	FDP_RIP.1	Subset residual information protection

Identification & Authentication (FIA)	FIA_AFL.1	Authentication failure handling
	FIA_IMA.1(Extended)	TOE Internal mutual authentication
	FIA_SOS.1	Verification of secrets
	FIA_UAU.2	User authentication before any action
	FIA_UAU.4	Single-use authentication mechanisms
	FIA_UAU.7	Protected authentication feedback
	FIA_UID.2	User identification 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)	ID and password management
	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)	Protection of stored TSF data
	FPT_TST.1	TSF self-testing
	FPT_FLS.1	Failure with preservation of secure state
TOE Access (FTA)	FTA_MCS.2	Per user attribute limitation on multiple concurrent sessions
	FTA_SSL3	Management of TSF-initiated sessions
	FTA_TSE.1	TOE session establishment
Secure Path/Channel (FTP)	FTP_ITC.1	Inter-TSF trusted channel

7.1 Security Audit

The TOE's security audit function consists of security alarms and audit data generation among the various functions of security audit (FAU)

7.1.1 Audit data generation

The following audit data is collected and stored by Manager and API.

- Security logs are generated by the security management function.
- Security logs are generated by the cryptographic operation function.
- Security logs include potential security violations, identification, authentication, TSF self-tests, and session termination.

If audit data is generated for changes in TSF data values, the modified TSF data values are also included in the audit data.

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

- Audit data generation time
- Audit data generation location
- Alert Level: INFO, ERROR
- Audit data type
- Audit result message

7.1.2 Potential security violation analysis and response actions

If a potential security violation is detected based on the generated audit data, the TSF shall take the actions described in [Table 18].

[Table 18] Response Actions to Potential Security Violations

Security Violation	Response Action
Accumulation of authentication failures as described in FIA_UAU.2	- If failed authentication attempts exceed the defined count (5 times), the account is locked for a period set by the administrator (default 10 minutes) - Send email to the administrator
Integrity violation and self-verification failure as described in FPT_TST.1	- Send email to the administrator
Exceeding predefined limits as described in FAU_STG.4	- Send email to the administrator
Audit storage saturation as described in FAU_STG.5	- Delete and record the oldest audit data - Send email to the administrator

7.1.3 Management of audit storage

The TOE takes the following measures when audit data exceeds the storage limit:

1) Notifies the administrator via email when the threshold defined by the administrator (default 90%) is reached.
2) If the audit data is saturated, deletes the oldest audit data, records new audit data, and notifies the administrator via email.

The percentage threshold indicates how much (%) of the total capacity of the audit data table space within the DBMS storing audit data is in use. For example, if a 90% threshold is set for a 100GB table space, it means the threshold is reached when 90GB of that table space is in use and 10GB is available.

7.1.4 Verification and review of audit data

The generated audit data is stored in the audit storage (refer to 7.1.1), and administrators verify and review the stored audit data through the screen interface (GUI) provided by the Manager.

The TSF provides a GUI function that allows authorized administrators to read audit data after accessing the Manager. When selection criteria values according to [Table 19] are entered for each audit data type, search results are displayed using an AND operation.

[Table 19] Audit Data Type and Selection Criteria

Audit Data Type	Selection Criteria (AND)	Search Method
TOE Log	Date and time	Search and sort (descending order based on audit data creation time)
	Log Level	
	Log occurrence type (Manager, API, Administrator)	
	Function Type	
	Result Message	

Related SFRs: FAU_ARP.1, FAU_GEN.1, FAU_SAA.1, FAU_SAR.1, FAU_SAR.3, FAU_STG.4, FAU_STG.5

7.2 Cryptographic Support

The TOE's cryptographic support function consists of authorized administrators establishing security policies, and, according to these security policies, generating and distributing cryptographic keys through a secure random bit generator. Cryptographic operations are processed via APIs within user applications. When an administrator calls a TOE process termination command, cryptographic keys stored in memory are destroyed

7.2.1 Cryptographic key generation and random number generation

The TOE uses interfaces provided by a validated cryptographic module for cryptographic key generation and random number generation, and the validated cryptographic module information is as follows:

Developer	AlphaBit Co., Ltd.
Cryptographic Module Name	AlphaCrypto V1.0
Validation Number	CM-265-2030.3
Validation Date	2025.03.07
Expiration Date	2030.03.07

The TOE generates TSF data encryption keys and user data encryption keys using the Hash_DRBG(SHA-256) algorithm provided by the validated cryptographic module for the following algorithms:

[Table 20] Data Encryption Key (DEK) Cryptographic Key Generation Algorithm and Key Size List

Standard List	Cryptographic Key Generation Algorithm	Cryptographic Key Length
TTAK.KO-12.0331	HASH_DRBG(SHA256)	128 bits
		192 bits
		256 bits

The TOE generates TSF data encryption keys using the Hash_DRBG(SHA-256) algorithm, public keys and private keys using the RSAES key pair generation algorithm, and signature keys and verification keys using the RSA-PSS key pair generation algorithm, according to the cryptographic generation algorithms and key lengths in [Table 20] to encrypt TSF data such as user information configuration and security policies.

[표 21] Cryptographic Key Generation Algorithm and Key Size List

Cryptographic Key Classification	Standard List	Cryptographic Key Generation Algorithm	Cryptographic Key Length
TSF data encryption key	TTAK.KO-12.0331	HASH_DRBG(SHA256)	256 bits
Public key/private key	KS X ISO/IEC 18033-2	RSAES(SHA256)	2048 bits
Signature key/verification key	KS X ISO/IEC 14888-2	RSA-PSS(SHA256)	2048 bits
Session key (symmetric key)	TTAK.KO-12.0331	HASH_DRBG(SHA256)	256 bits

The TOE uses a Hash_DRBG (SHA-256) random number generator to generate cryptographic keys for the ARIA block cipher algorithm. The length of the key generated by the random number generator is generally 256 bits and is generated with the length of the input parameter (multiple of 8, positive). It uses an RSAES key pair generator to generate key pairs for RSAES. It uses an RSA-PSS key pair generator to generate key pairs for RSA-PSS. The length of the keys generated by each key pair generator is 2048 bits. If a length corresponding to each cryptographic algorithm is input, a key of that length is generated.

the validated cryptographic module information is as follows

Developer	AlphaBit Co., Ltd.
Cryptographic Module Name	AlphaCrypto V1.0
Validation Number	CM-265-2030.3
Validation Date	2025.03.07
Expiration Date	2030.03.07

The TOE's entropy source mechanism for collecting noise source output and utilizing the returned entropy output to configure the seed used as input for the DRBG is described as follows:

- List of Noise Sources Included in the Entropy Source and Entropy Provided by Each Noise Source

No.	Name	Bytes	Repetition Count	Entropy	Remarks
1	gettimeofday	8	1	-	Information about the current time, used as an auxiliary noise source
2	/dev/urandom	1	45	6.825867	Rocky Linux random number generator, used as a main noise source concatenated after the auxiliary noise source.

- Noise Source Output Collection and Composition of the Entropy Source

The total collected noise comprises 68 bytes, consisting of 8 bytes of auxiliary noise and 60 bytes of main noise. The auxiliary noise is collected via the gettimeofday system function. The main noise of 60 bytes is collected using /dev/urandom. A health test is performed on the main noise, and an additional 15 bytes of noise are collected for the Adaptive Proportion Test. After passing the noise source health test, the actually used main noise is 45 bytes.

- Performs Noise Source Health Test

The noise source health test is performed when collecting noise sources, and the details regarding the health test are as follows:

- Repetition Count Test (RCT)

Method	- Observe the number of occurrences B of the same value collected continuously. - If B reaches cutoff C, a severe error occurs.
Cutoff C	$C = \lceil 1 + (-\log_2 \alpha / \hat{H}) \rceil$ ($\alpha = 2^{-20}$, $\hat{H}$ = Entropy of Noise Source)
Pseudo-code	Input: Dataset S = (s_1, ..., s_L), Dataset size L, Cutoff value C Algorithm: 1) Let variable t ← s_1 2) Initialize counter variable count to 1. (count ← 1) 3) For i from 2 to L: 3.1) If s_i = t then 3.1.1) count ← count + 1 3.1.2) If count = C, output True (Error exists) 3.2) If $s_i \neq$ t then 3.2.1) t ← s_i 3.2.2) count ← 1 4) Output False Output: Error existence T/F

- Adaptive Proportion Test (APT)

Method	- Observe the number of occurrences B of the same value as the first noise source value within a specific range W. - If B reaches cutoff C, a simple error occurs. Cutoff C Pseudo-code
Cutoff C	C = 1 + CRITBINOM(W, 2^(–$\hat{H}$), 1 – α) ($\alpha = 2^{-20}$, $\hat{H}$ = Entropy of Noise Source)
Pseudo-code	Input: Dataset S = (s_1, ..., s_W), Dataset size W, Cutoff value C Algorithm: 1) Let t ← s_1 2) Initialize counter variable count to 1. (count ← 1) 3) For i from 2 to W: 3.1) If s_i = t then count ← count + 1 4) If count > C, output True (Error exists) 5) If count ≤ C, output False Output: Error existence T/F

- Performs Conditioning Process (Optional)

A conditioning process is performed through the SHA256 hash algorithm with entropy consisting of 8 bytes of additional noise source and 45 bytes of main noise source as input.

- Entropy Source Output Configuration

A 32-byte entropy source is configured through the conditioning process.

- Seed Configuration and Seed Entropy (bits)

According to "6.1 Overview" of the standard [Hash Function Based Deterministic Random Bit Generator – Part 1: General TTAK.KO-12.0331-Part1], it consists of a 32-byte entropy source and 24 bytes of /dev/urandom value, and has an entropy value of 256 bits.

7.2.2 Cryptographic key distribution

The cryptographic key distribution methods used in the TOE are as follows: Session keys are distributed between Manager and API during mutual authentication and secure channel establishment. When the Manager distributes a session key to the API, it is encrypted and distributed with the identified API's public key. After the secure channel is established, user data encryption keys are distributed to the API. These encryption keys are encrypted and distributed using the session key distributed during secure channel establishment.

The session key distribution method for internal TOE mutual authentication and protection of transmitted data is as follows:

[Table 22] Cryptographic Key Distribution Algorithm and Key Size List

Classification	Cryptographic Algorithm	Cryptographic Key Length	Reference Standard
Public key cryptography	RSAES(SHA-256)	Public key 2048 bits	KS X ISO/IEC 11770-3
Cryptographic Key Distribution Method			
To distribute cryptographic keys between Manager and API, the entity transmitting the cryptographic key (Manager) encrypts and distributes it using the recipient's (API) public key.			

7.2.3 Cryptographic key derivation

Since the validated cryptographic module does not provide a key derivation interface, the TOE implements the PBKDF2(SHA-256) algorithm specified in the TTAK.KO-12.0334 standard to derive cryptographic keys.

The cryptographic key derivation method for protecting TSF data used in the TOE is as follows:

[Table 23] Key Encryption Key (KEK) Key Derivation Algorithm and Key Size List

Standard List	Key Derivation Algorithm	Cryptographic Key

		Length
TTAK.KO-12.0334	PBKDF2(SHA-256)	256 bits

7.2.4 Cryptographic key destruction

KEK in memory is deleted when the administrator calls the Manager process termination command. DEK stored in memory is destroyed before the use of cryptographic function interfaces provided by the TOE ends. Session keys used for communication between TOE components are released from memory and destroyed when communication ends. KEKs generated by derivation from passwords are not stored, but are generated when the Manager runs and stored in memory.

The cryptographic key destruction method is as follows:

- Overwrite cryptographic key memory with "0x00" value 3 times.

[Table 24] Cryptographic Key Type-Specific Destruction Method

Classification	Cryptographic Key Type	Destruction Method
Manager	KEK	Overwrite KEK memory with "0x00" value 3 times
	TSF data encryption key	Overwrite TSF data encryption key memory with "0x00" value 3 times
	Session key	Overwrite session key memory with "0x00" value 3 times
API	KEK	*Overwrite KEK memory with "0x00" value 3 times
	DEK	Overwrite DEK memory with "0x00" value 3 times
	TSF data encryption key	Overwrite TSF data encryption key memory with "0x00" value 3 times
	Session key	Overwrite session key memory with "0x00" value 3 times
	User data encryption key	Overwrite user data encryption key memory with "0x00" value 3 times

7.2.5 Cryptographic operations

The TOE's cryptographic operations are divided into cryptographic operation functions for preventing leakage and securing security policies, and cryptographic operation functions for user DB data.

The validated cryptographic module information used in the TOE is as follows

Developer	AlphaBit Co., Ltd.
Cryptographic Module Name	AlphaCrypto V1.0

Validation Number	CM-265-2030.3
Validation Date	2025.03.07
Expiration Date	2030.03.07

The cryptographic algorithms and key sizes used for user data cryptographic operations are as follows.

[Table 25] User Data Cryptographic Standards and Algorithms

Standard	Cryptographic Algorithm	Cryptographic Key Length	Operation Mode	Function List
KS X 1213-1	ARIA	128, 192, 256	CBC	Data encryption/decryption
KS X ISO/IEC 10118-3	SHA-256,	None	None	Data one-way encryption

The cryptographic algorithms and key sizes used for TSF data cryptographic operations are as follows.

[Table 26] TSF Cryptographic Standards and Algorithms

Component	Standard	Cryptographic Algorithm	Cryptographic Key Length	Operation Mode	Function List
Manager	KS X 1213-1	ARIA	256	CBC	User data encryption key encryption/decryption
	KS X 1213-1	ARIA	256	CBC	Configuration information encryption/decryption
	KS X 1213-1	ARIA	256	CBC	Session data encryption/decryption
	KS X ISO/IEC 10118-3	SHA-256	None	None	Password hash value generation
	PKCS#1 v2.1	RSA-PSS	2048	None	Signature/verification during mutual authentication

	PKCS#1 v2.1	RSAES	2048	None	Key distribution encryption/decryption
	RFC2898 (PKCS#5v2.1)	PBKDF2 (SHA-256)	256	None	Password-based key derivation
API	KS X 1213-1	ARIA	256	CBC	User data encryption key encryption/decryption
	KS X 1213-1	ARIA	256	CBC	Session data encryption/decryption
	KS X ISO/IEC 10118-3	SHA-256	None	None	Data one-way encryption
	PKCS#1 v2.1	RSA-PSS	2048	None	Signature/verification during mutual authentication
	PKCS#1 v2.1	RSAES	2048	None	Key distribution encryption/decryption
	RFC2898 (PKCS#5v2.1)	PBKDF2 (SHA-256)	256	None	Password-based key derivation

The TOE uses the following cryptographic key generation methods.

Classification	Cryptographic Key Type	Generation Method
Manager	KEK	Generated via PBKDF2 algorithm using the password input at initial startup and a 16-byte random number generated by the validated cryptographic module's Hash_DRBG(SHA-256) algorithm as salt.
	TSF data encryption key	TSF data encryption key is generated as a 32-byte random number using the validated cryptographic module's Hash_DRBG(SHA-256) algorithm. IV for TSF data encryption is generated as a 16-byte random number using the validated cryptographic module's Hash_DRBG(SHA-256) algorithm.
	Session key	A 32-byte session key and a 16-byte IV are generated using the validated cryptographic module's

		Hash_DRBG(SHA-256) algorithm.
	User data encryption key	A 32-byte user data encryption key and a 16-byte IV are generated using the validated cryptographic module's Hash_DRBG(SHA-256) algorithm.
API	KEK	KEK is derived via the PBKDF2 algorithm using the password information (password, Salt, IV) input during API public key pair generation. The input password information is distributed as an encoded file (master_key) using the Manager's self-encoding method, and the password information is obtained by decoding it each time it is used.
	TSF data encryption key	TSF data encryption key and IV are generated as 32-byte and 16-byte random numbers, respectively, using the validated cryptographic module's Hash_DRBG(SHA-256) algorithm.

The TOE uses the following cryptographic key distribution methods

Cryptographic Key Type	Distribution Method
Session key	Session key is encrypted with the API's public key and distributed.
User data encryption key	User data encryption key is distributed using the distributed session key.

The TOE performs cryptographic key destruction as follows:

Classification	Cryptographic Key Type	Destruction Method
Manager	KEK	Overwrite KEK memory with "0x00" value 3 times
	TSF data encryption key	Overwrite TSF data encryption key memory with "0x00" value 3 times
	Session key	Overwrite session key memory with "0x00" value 3 times
	User data encryption key	Overwrite user data encryption key memory with "0x00" value 3 times
API	KEK	Overwrite KEK memory with "0x00" value 3 times
	TSF data encryption key	Overwrite TSF data encryption key memory with "0x00" value 3 times
	Session key	Overwrite session key memory with "0x00" value 3 times

	User data encryption key	Overwrite user data encryption key memory with "0x00" value 3 times

Related SFRs: FCS_CKM.1(1), FCS_CKM.1(2), FCS_CKM.2, FCS_CKM.6, FCS_COP.1(1), FCS_COP.1(2), FCS_RBG.1, FCS_RBG.3

7.3 User Data Protection

The TOE provides column encryption/decryption functionality to protect user data. Additionally, the TOE ensures the security of important information residing in memory by performing memory initialization of key information when the TOE process terminates.

The TOE uses interfaces provided by a validated cryptographic module for encryption/decryption functions to protect user data, and the validated cryptographic module information is as follows:

Developer	AlphaBit Co., Ltd.
Cryptographic Module Name	AlphaCrypto V1.0
Validation Number	CM-265-2030.3
Validation Date	2025.03.07
Expiration Date	2030.03.07

The API provides user data encryption and decryption functions by calling encryption interfaces (e.g., encryption/decryption API) from user applications.

- API

Authorized administrators establish security policies, such as cryptographic key generation, in the Manager, and then distribute the cryptographic keys and security policies to the API. After performing an authentication process with the Manager, the API receives the security policies, stores them in memory, and calls the encryption/decryption interface to perform user data encryption.

[Table 27] shows the user data cryptographic standards and algorithms for encryption/decryption in the API.

[Table 27] User Data Cryptographic Standards and Algorithms

Standard List	Cryptographic Algorithm	Cryptographic Key Length	Operation Mode	Cryptographic Operation List
KS X 1213-1	ARIA	128, 192, 256	CBC	User data encryption/decryption

Related SFRs: FDP_UDE.1, FDP_RIP.1

7.4 Identification and Authentication

The TOE provides mutual authentication between TOE components and administrator identification and authentication functions when security administrators connect to the Manager through a management tool.

7.4.1 Administrator identification and authentication

Security administrators must modify the administrator account (admin) password and register allowed IP addresses when installing the Manager. The management tool performs security administrator authentication using an ID and password before the security administrator performs security management functions. When performing security administrator identification and authentication, the management tool masks the password entered by the security administrator with an asterisk (*) so that it is not displayed on the screen, and provides only an authentication failure message "Invalid Login Info" upon authentication failure. If authentication fails for a defined number of consecutive times (fixed: 5 times), access to the account is blocked for 5 to 10 minutes (administrator setting), an audit record for the authentication failure is stored, and a warning email is sent to the security administrator.

The TOE provides a verification mechanism that meets the administrator password generation rules for administrator authentication and password changes.

- Password length must be between 9 and 20 characters, and allowed characters are limited to uppercase letters, lowercase letters, special characters, and numbers. In addition, it must include at least one uppercase letter, one lowercase letter, one special character, and one number.
- The same password as the user account (ID), passwords with 3 or more consecutive identical characters/numbers, passwords with 3 or more consecutive characters or numbers entered sequentially on the keyboard, and the immediately preceding password cannot be used.

Furthermore, to ensure the uniqueness of sessions used when a security administrator accesses the management tool, the administrator provides functionality to prevent reuse of authentication data by using random number information generated through a random number generator.

7.4.2 TOE mutual authentication

During communication between TOE components, mutual authentication is performed in real-time through digital signature verification using unique public key/private key pairs issued between the

Manager and API. The Manager and API public key/private key pairs used for mutual authentication are generated through the Manager. The Manager stores the public key hash value of the generated API. When performing mutual authentication, the API transmits the public key hash value, integrity verification value, and signature value generated with the Manager's public key to the Manager. The Manager identifies the API public key hash value by comparing it with the stored hash value, and performs integrity verification and signature value verification.

Related SFRs: FIA_AFL.1, FIA_IMA.1, FIA_SOS.1, FIA_UAU.2, FIA_UAU.4, FIA_UAU.7, FIA_UID.2

7.5 Security Management

7.5.1 Management of security functions behavior

The TOE provides security management functions only when administrator identification and authentication are successfully performed. Security administrators can access the security management interface only through a secure channel. The list of security functions managed by authorized administrators is shown in the following table.

Security Management Function	Management Action
Initial setup and modification of administrator password	Modify
Setting and modification of audit record storage threshold	View, Modify
Modification of authentication failure count and administrator account deactivation time	View, Modify
Viewing of audit records	View
Setting and modification of cryptographic policies	Create, View, Modify, Delete
Setting and modification of cryptographic keys	Create, View, Modify

7.5.2 TSF data management

Only security administrators successfully authenticated to the TOE can manage TSF data. The list of TSF data managed by authorized administrators is shown in the following table.

TSF Data List	Management
Administrator account information	View
Administrator access IP	View, Modify
SMTP access account information	View, Modify
Setting of audit information storage threshold	View, Modify

Encryption policy	Create, View, Modify, Delete
Cryptographic key (User data encryption)	Create, View, Modify
Audit information	View

7.5.3 Security roles and ID and password management

The TOE generates the security administrator's ID and password when the Manager is installed, and identifies and authenticates the security administrator through them. The security roles provided by the TOE are limited to security administrators, and password changes can only be performed by security administrators through the management tool. The TOE defines password combination rules for security administrator identification and authentication and provides password change functionality.

Related SFRs: FMT_MOF.1, FMT_MTD.1, FMT_PWD.1, FMT_SMF.1, FMT_SMR.1

7.6 Protection of the TSF

7.6.1 Basic internal TSF data transfer protection

When TSF data is transmitted between separated components of the TOE, the secure cryptographic communication protocol TLS v1.2 is used to prevent disclosure and manipulation of data.

Detailed algorithm information supported within the TLS protocol (TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384, TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384) is as follows:

- Key Exchange : ECDHE
- Authentication : RSA 3072-bits or ECDSA
- Encryption : AES – 256 - GCM
- Hash : SHA384

7.6.2 Basic protection of stored TSF data

The TOE protects stored TSF data from unauthorized disclosure and manipulation by encrypting, storing, and managing protected TSF data. Information that requires encryption includes administrator passwords, TOE configuration information (DB storage information, SMTP configuration information, Manager configuration files), etc. Administrator passwords are

encrypted with Salt Hash using the SHA-256 algorithm, and TOE configuration values are encrypted with ARIA-CBC-256. The Manager's DEK file and Manager's private key file are also encrypted in the same way.

The cryptographic algorithms applied to the protected TSF data are as follows:

[Table 28] TSF Data Storage Cryptographic Algorithm

Classification	TSF Data	Crypto Algorithm	Remarks
TOE Server	Password used by TOE for user identification and authentication	SHA256	During hashing, add 128 bits generated using Hash_DRBG(SHA256) to the password, apply an iteration count of 1000, and store in the DBMS interworking with the TOE (protected from unauthorized user access using DBMS identification and authentication functions).
	Original hash value for integrity verification target	ARIA-256(CBC)	Encrypted with TSF data encryption DEK and stored in file system (/asm/integrity/).
	DBMS access information (ID/PW)	ARIA-256(CBC)	Encrypted with TSF data encryption DEK and stored in file system (/asm/conf/application.properties).
	SMTP settings (ID, PW)	ARIA-256(CBC)	Encrypted with TSF data encryption DEK and stored in DBMS (protected from unauthorized user access using DBMS identification and authentication functions)).
	KEK	-	Store 128-bit Salt generated using Hash_DRBG(SHA256) in file system (/asm/keystore/salt). KEK is not stored, but loaded into memory and destroyed from memory when the TOE server terminates.
	User data encryption DEK	ARIA-256(CBC)	Encrypted with KEK, base64 encoded, and stored in DBMS (protected from unauthorized user access using DBMS identification and authentication functions)).
	TSF data encryption DEK	ARIA-256(CBC)	Encrypted with KEK and stored in file system (/asm/keystore/dekServer).
	RSAES private key	ARIA-	Encrypted with KEK and stored in file

		256(CBC)	system (/asm/keystore/server_key_enc.pri).
	RSA-PSS verification key	ARIA-256(CBC)	Encrypted with KEK and stored in file system (/asm/keystore/server_key_sig.pri).
TOE API	Original hash value for integrity verification target	ARIA-256(CBC)	Encrypted with DEK and stored in file system (alpha/smart/sapi_root/hashList).
	KEK	-	Store password, Salt, and IV generated using Hash_DRBG(SHA256) and entered by the administrator through self-encoding (/api/sapi_root/keystore/master_key). KEK is not stored, but loaded into memory and destroyed from memory when the API terminates.
	TSF data encryption DEK	ARIA-256(CBC)	Encrypted with KEK and stored in file system (/api/sapi_root/keystore/conf.key).
	RSAES private key	ARIA-256(CBC)	Encrypted with KEK and stored in file system (/api/sapi_root/keystore/client_key1_enc.pri).
	RSA-PSS verification key	ARIA-256(CBC)	Encrypted with KEK and stored in file system (/api/sapi_root/keystore/client_key1_sig.pri).

7.6.3 TSF self-testing

TSF testing consists of process self-testing and integrity verification. If self-testing or integrity verification fails, a warning email is sent to the email address set by the security administrator. The TOE periodically checks if processes between TOE components are running normally at startup and every hour.

[Table 29] TOE Self-Test Targets

Self-Test Item	Test Content
Execution Process Manager	Performs self-test at startup and generates audit logs. Creates a test account and verifies that data is normally entered into the database. Creates/deletes test keys and encryption policies to verify normal operation of KCMVP modules used in cryptographic operations. Records audit log and stops process if self-test fails.

	Monitors if processes are running normally every hour during operation.

In addition, the TOE performs integrity verification tests (validated cryptographic module, executable/configuration file verification) at startup, periodically (every hour), and upon request by an authorized administrator.

[Table 30] TOE Integrity Test Targets

Classification	Integrity Test Item	Test Content
Manager	- Manager Module libAlphaCrypto1.0.so libAlphaCSP.so libAlphaCSP_JNI.so ap-DBGuard-asm-2.1.0.3.jar rt.jar	During installation, the Hash (SHA-256) value of each generated file is stored. During execution, the Hash value of the module is verified against the stored value. This process is then performed periodically every hour.
	- Configuration File application.properties server_key_sig.pri server_key_sig.pub server_key_enc.pri server_key_enc.pub	During installation, the Hash (SHA-256) value of each generated file is stored. During execution, the Hash value of the module is verified against the stored value. This process is then performed periodically every hour.
API	- API Module libAlphaCrypto1.0.so libAlphaCSP.so libAlphaCSP_JNI.so ap-DBGuard-api-2.1.0.3.jar ap-DBGuard-jni-2.1.0.3.jar ap-DBGuard-common-2.1.0.3.jar rt.jar	During installation, the Hash (SHA-256) value of each generated file is stored. During execution, the Hash value of the module is verified against the stored value. This is performed only during execution and not periodically.
	- Configuration File client_key1_sig.pri client_key1_sig.pub client_key1_enc.pri client_key1_enc.pub	During installation, the Hash (SHA-256) value of each generated file is stored. During execution, the Hash value of the module is verified against the

	server_key_sig.pub server_key_enc.pub sapi.cfg conf.key master_key	stored value. This is performed only during execution and not periodically.

If the integrity verification of a target fails, the authorized administrator is notified by email. As there is a possibility that the TOE component's information has been tampered with, manual recovery of the TOE, such as reinstallation or restart, must be performed. Furthermore, if a random number health test fails, leading to a severe error and application operation termination, the affected TOE must be restarted to restore the application to a normal state

Related SFRs: FPT_ITT.1, FPT_PST.1, FPT_TST.1, FPT_FLS.1

7.7 TOE Access

The TOE provides functionality to restrict administrator access based on the access IP for the same TSF administrator. The accessible IP is set during the initial installation of the TOE. The TOE does not restrict the number of accessible IPs by default. The maximum number of concurrent sessions is also limited to 1. It manages the list of active sessions based on session ID, and if an administrator logs in via a different IP, the existing session is terminated. If inactivity persists for a certain period (10 minutes), the TOE checks the time, requests session termination from the WAS, and terminates the session with WAS support.

Related SFRs: FTA_MCS.2, FTA_SSL.3, FTA_TSE.1

7.8 Secure Path/Channel

The TOE supports the secure cryptographic communication protocol TLS v1.2 to protect transmitted data when performing email transmission functions using SMTP.

Detailed algorithm information supported within the TLS protocol (TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) is as follows:

- Key Exchange : ECDHE
- Authentication : RSA 3072-bits
- Encryption : AES – 256 - GCM
- Hash : SHA384

Related SFRs: FTP_ITC.1