Petra Cipher V4.0-ASE(Security Target)-V1.4

Petra Cipher V4.0

2024.10.08

SINSIWAY Co., Ltd. R&D Center

< Document History >

Version	Revision	Date	Author
V1.0	Initial registration	2024.05.02	Hyewon Lee
V1.1	Reflection of observations	2024.07.19	Hyewon Lee
V1.2	Correction of Typos and Awkward Sentences	2024.08.05	Hyewon Lee
V1.3	Change of TOE detailed version in the document (V4.0.0.2)	2024.09.23	Hyewon Lee
V1.4	Preparative Procedure version edit	2024.10.08	Hyewon Lee

< Table of Contents >

1. ST Introduction

This Security Target (ST) defines the security functional requirements of Petra Cipher V4.0, the security functions to satisfy the security functional requirements, and assurance requirements for secure assurance.

1.1. ST reference

Classification	Description
Title	Petra Cipher V4.0-ASE(Security Target)
ST Version	V1.4
Developer	SINSIWAY Co., Ltd. R&D Center
Publication Date	October 08, 2024
Common Criteria	Common Criteria for Information Technology Security Evaluation Version 3.1 Revision 5 ▪ Common Criteria for Information Technology Security Evaluation Part 1: Introduction and General Model, V3.1 r5 (CCMB-2017-04-001, April 2017) ▪ Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Components, V3.1 r5 (CCMB-2017-04-002, April 2017) ▪ Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Components, V3.1 r5 (CCMB-2017-04-003, April 2017)
Common Criteria Version	V3.1 R5
Evaluation Assurance Level	EAL1+ (ATE_FUN.1)
Keywords	DB encryption, encryption, database

[Table 1-1] ST reference

1.2. TOE reference

Classification	Description	
TOE	Petra Cipher V4.0	
Version Detail	V4.0.0.2	
TOE Component	Petra Cipher Key Server	Petra Cipher Key Server V4.0.0.2

		Petra Cipher DB Agent	Petra Cipher DB Agent V4.0.0.2
		Petra Cipher API Agent	Petra Cipher API Agent V4.0.0.2
Guidance Document	Operational Guidance	Petra Cipher V4.0-OPE(Operational Guidance)-V1.2	
	Preparative Procedure	Petra Cipher V4.0-PRE(Preparative Procedure)-V1.3	
	Developer Guide	Petra Cipher V4.0-API(Developer Guide)-V1.1	
Developer		SINSIWAY Co., Ltd. R&D Center	

[Table 1-2] TOE reference

1.3. Overview of the TOE

Petra Cipher V4.0 (hereinafter referred to as "TOE") is a database encryption product developed by Shinsegae Co., Ltd. (hereinafter referred to as "DB"). The TOE performs functions to prevent unauthorized exposure of information by encrypting the database.

The target of TOE encryption is the database managed by the Database Management System (DBMS) in the organization's operational environment. In this security target specification, all data before and after being encrypted and stored in the DB is defined as user data. Depending on the security policy of the organization operating the TOE, part or all of the user data may be subject to encryption.

1.3.1. TOE Type and Scope

The TOE is provided in the form of software and offers column-level encryption and decryption functions for user data. The TOE is classified into two types based on its operation method: the "Plug-in method" and the "API method," and the TOE supports both methods.

The components of the TOE include the Petra Cipher Key Server, Petra Cipher DB Agent, and Petra Cipher API Agent.

- Petra Cipher Key Server
  Petra Cipher Key Server is installed on the Management Server, and manages a master key used to protect an encryption key and a key for encryption/decryption. It also manages

decryption authority and key request logs. It provides a user interface (GUI) that allows an authorized administrator to access Petra Cipher Key Server via an internet web browser and to perform security management functions such as policy establishment.

- Petra Cipher DB Agent
  Petra Cipher DB Agent is installed on the Database Server. It generates an encryption management account inside the DB to be protected, and then installs a necessary package, thereby providing DB API for using actual encryption/decryption functions.
- Petra Cipher API Agent
  Petra Cipher API Agent is installed on the Application Server, and provides API for using user data encryption/decryption functions delivered to the Application Server. Petra Cipher API Agent provides C and JAVA API.

1.3.2. Usage and major security features of the TOE

The TOE is used to encrypt user data according to the policy set by the authorized administrator to prevent the unauthorized disclosure of the information required to be protected. The TOE uses a validated cryptographic module whose security has been validated by Korea Cryptographic Module Validation Program (KCMVP).

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

Data Encryption Key (DEK) used to encrypt/decrypt user data is protected by the encryption with Key Encryption Key (KEK). DEK is also used to protect the stored TSF data and communication among TOE component, and is performed by using a cryptographic algorithm approved in a cryptographic module

whose security and implementation conformance have been validated by Korea Cryptographic Module Validation Program (KCMVP).

1.3.3. TOE operational environment

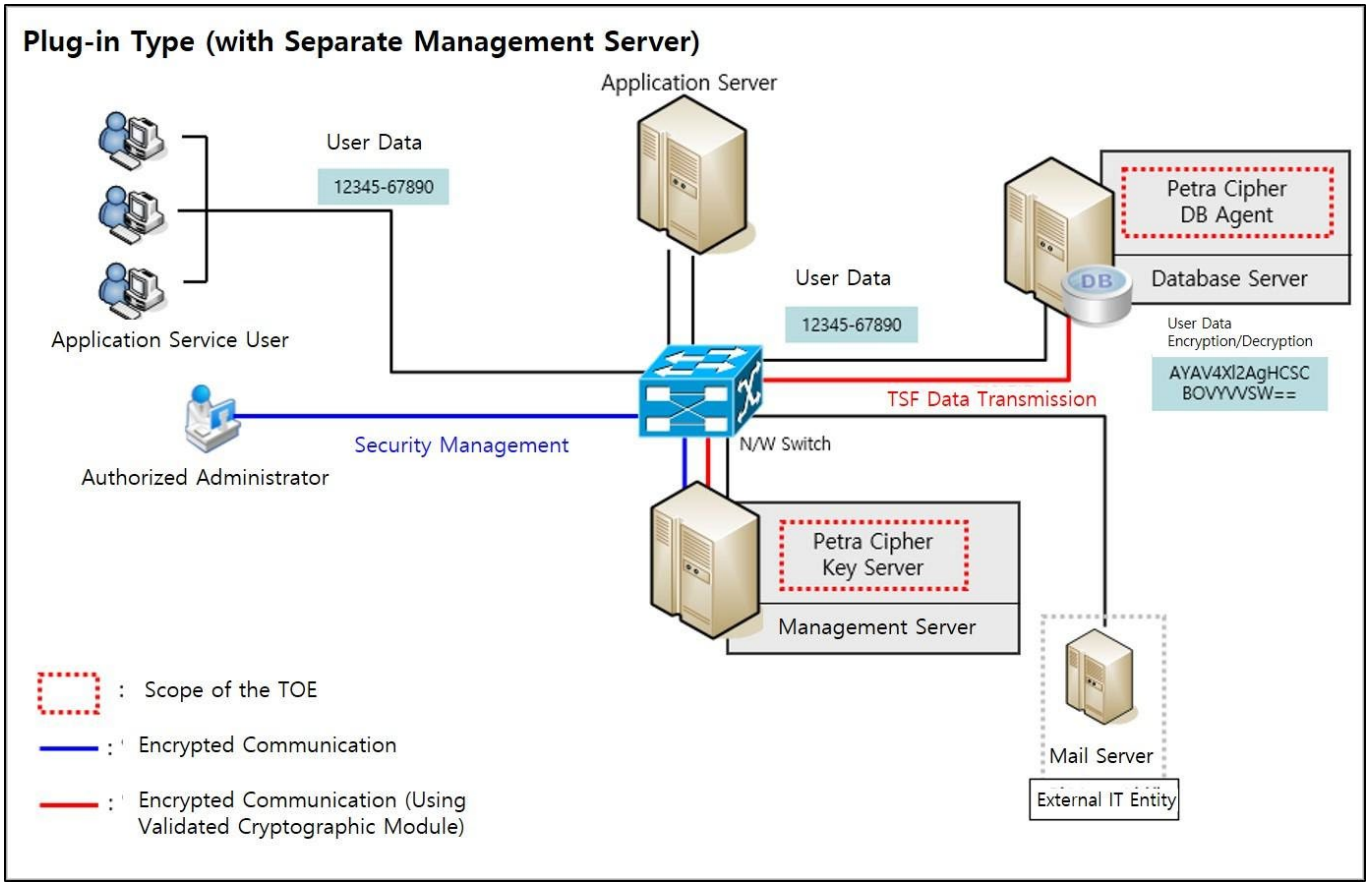
The TOE operational environment is classified into two: plug-in type and API type. Both types are operated with the Management Server and the Database Server where the DB to be secured is installed separated.

1.3.3.1. Plug-in type

[Figure 1-1] below shows the operational environment of the plug-in type offered by the TOE.

Petra Cipher DB Agent, which is installed on the Database Server, encrypts user data received from the Application Server before they are stored in the DB according to the policies established by the authorized administrator, and decrypts encrypted user data transmitted from the Database Server to the Application Server.

In addition, Petra Cipher DB Agent collectively encrypts user data that have already been stored in the Database Server through Petra Cipher Key Server, and decrypts the encrypted user data whenever there is a request from an application service user.



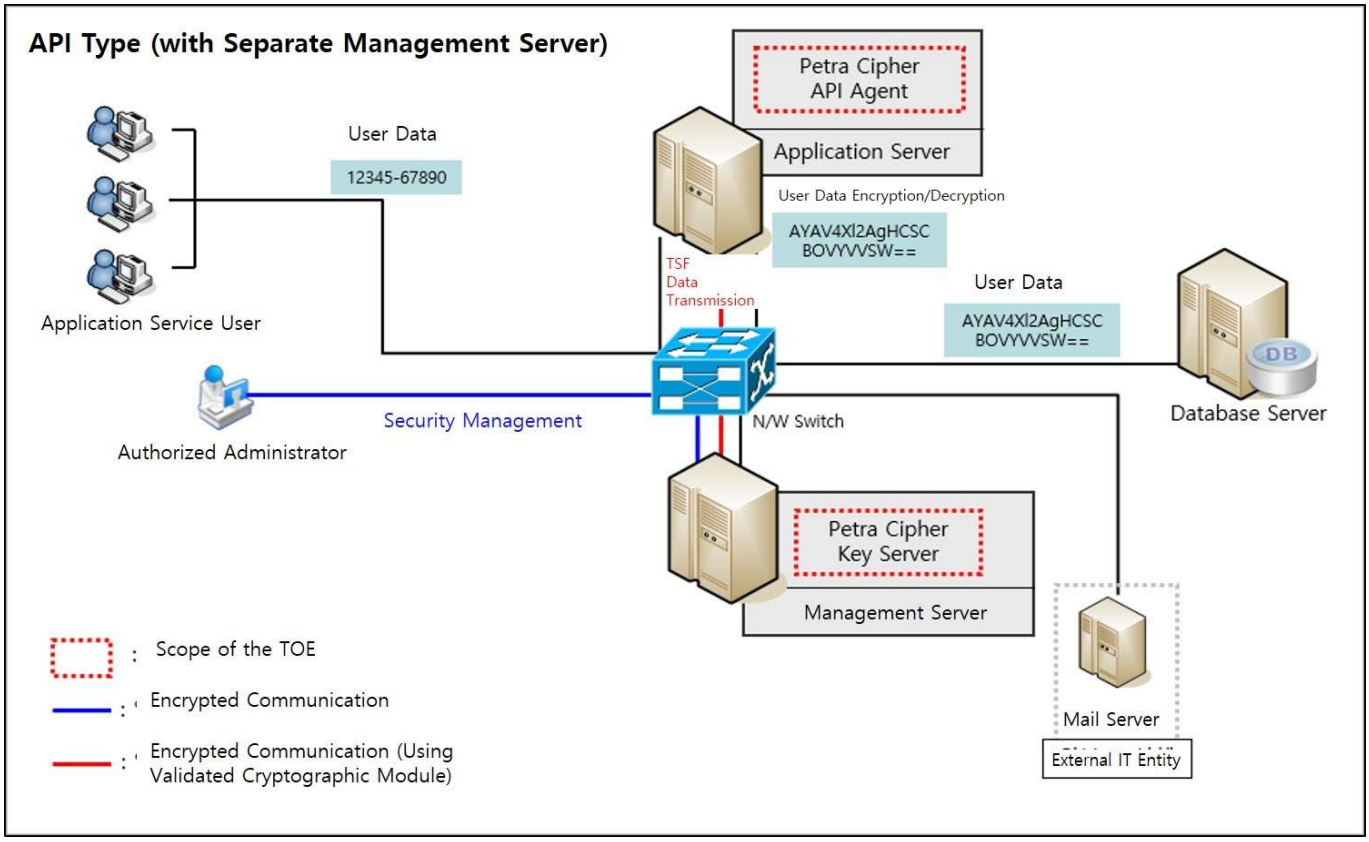
[Figure 1-1] Operational environment of plug-in type

The authorized administrator can access Petra Cipher Key Server inside the Management Server via a web browser, and encrypts/decrypts user data and performs the security management according to the scope of the encryption required by the organizational security policy.

1.3.3.2. API type

[Figure 1-2] below shows the operational environment of the API type offered by the TOE.

The application that is installed on the Application Server and provides application services has been developed as API provided by Petra Cipher API Agent in order to use cryptographic functions of the TOE. Petra Cipher API Agent is installed on the Application Server and encrypts/decrypts user data according to the polices established by the authorized administrator. User data entered by an application service user are encrypted by Petra Cipher API Agent installed on the Application Server, and sent to the Database Server. Encrypted user data received from the Database Server are decrypted by Petra Cipher API Agent installed on the Application Server, and sent to the application service user.



[Figure 1-2] Operational environment of API type

The authorized administrator can access Petra Cipher Key Server inside the Management Server via a web browser, and encrypts/decrypts user data and performs the security management according to the scope of the encryption required by the organizational security policy.

The communication among the TOE components is based on the encrypted communication using the approved cryptographic algorithm of the validated cryptographic module. When the authorized administrator accesses the Management Server through a web browser, a secure path shall be generated and used for the communication.

Classification	Description
Cryptographic Module Name	KLIB V3.0
Developer	Korea University
Validation Date	August 6, 2021
Validation Level	General level: Security Level 1 Level by Item - Cryptographic module specification Level 1 - Cryptographic module interface Level 1 - Roles, services and authentication Level 1

	- Software/firmware Security Level 1 - Operational environment Level 1 - Physical security N/A - Non-invasive security N/A - Critical security parameter management Level 1 - Self-test Level 1 - Life cycle assurance Level 1 - Response to other attacks Level 1
Validation No.	CM-189-2026.8

[Table 1-3] Information on the validated cryptographic module used in the TOE

1.3.4. Identification of non-TOE hardware/software/firmware

[Figure 1-4] below describes specifications of non-TOE hardware/software for the operation of the TOE.

Classification		TOE Component		
		Petra Cipher Key Server	Petra Cipher DB Agent	Petra Cipher API Agent
Hardware	CPU	Intel(R) Core(TM) i5-4250U CPU @ 1.30GHz or higher		
	RAM	4GB or higher		
	HDD	Space required for installation of TOE 50 GB or higher	Space required for installation of TOE 1 GB or higher	Space required for installation of TOE 1 GB or higher
	NIC	NIC: 100/1000 Mbps Ethernet Port 1 unit or more		
Software		OS : Rocky-Linux-9 (Kernel 5.14.0) 64 bit		
		- apache-tomcat 9.0.93 - openJDK 13.0.2	- DBMS (DB to be protected): Oracle 19c(19.3.0.0.0)	- openJDK 13 : 13.0.2

[Table 1-4] Hardware/software environment for the operation of the TOE

- Further information on Petra Cipher Key Server software
  - Apache-Tomcat is software to provide the web access environment of Petra Cipher Key Server.
  - OpenJDK is software to operate Apache-Tomcat.
- Further information on Petra Cipher API Agent software
  - OpenJDK is software to operate JAVA Application and execute JAVA API.

[Table1-5] below describes the external entity necessary for the operation of the TOE.

Classification	Use

Mail Server	Mail Server to send an alarm to an email set by the authorized administrator

[Table 1-5] Non-TOE software environment

[Table1-6] below shows the environment requirements for the operation of the administrator PC.

Classification	Item	Requirements
Software	Web browser	Chrome 129

[Table 1-6] Environment requirements for the operation of the administrator PC

1.4. TOE description

This section describes the physical/logical scope and boundary of the TOE.

1.4.1. Physical scope and boundary of the TOE

The TOE consists of Petra Cipher Key Server, Petra Cipher DB Agent, Petra Cipher API Agent software, along with an operator's manual, preparation procedures, and a developer's guide. Hardware and OS required for TOE operation, openJDK, Apache-Tomcat, and the target DBMS are excluded from the evaluation scope.

Classification	Description		Type	Delivery Method
TOE	Petra Cipher V4.0		-	-
Version Detail	V4.0.0.2		-	-
TOE Component	Petra Cipher Key Server	Petra Cipher Key Server V4.0.0.2 File Name: installer-Petra_Cipher_V4.0.0.2-linux-64bit.tar.gz	S/W	The CD is placed in a CD case, and then sealed with a label and delivered
	Petra Cipher DB Agent	Petra Cipher DB Agent V4.0.0.2 File Name: dbagent-Petra_Cipher_V4.0.0.2-linux-64bit.tar.gz	S/W	
	Petra Cipher API Agent	Petra Cipher API Agent V4.0.0.2 File Name: apiagent-Petra_Cipher_V4.0.0.2-linux-64bit.tar.gz	S/W	
Document	Operational guideline	Petra Cipher V4.0-OPE(Operational Guideline)-V1.2 File Name: Petra Cipher V4.0-OPE(Operational Guideline)-V1.2.pdf	PDF Document	
	Preparative Procedure	Petra Cipher V4.0-PRE(Preparative Procedure)-V1.3 File Name: Petra Cipher V4.0-PRE(Preparative Procedure)-V1.3.pdf		
	Developer Guide	Petra Cipher V4.0-API(Developer Guide)-V1.1 File Name: Petra Cipher V4.0-API(Developer Guide)-V1.1.pdf		

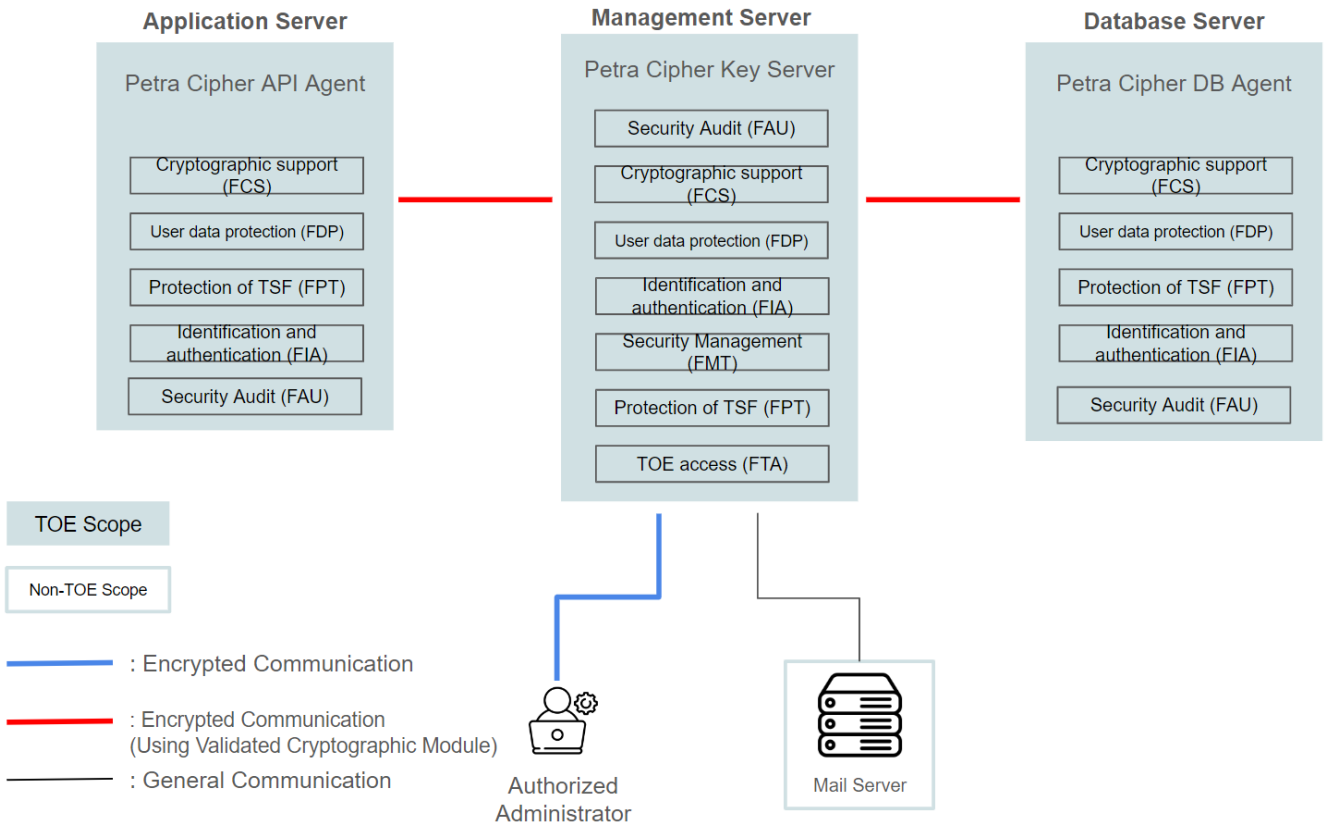
[Table 1-7] Physical scope of the TOE

The communication among separate components of the TOE, such as the communication among

Petra Cipher Key Server, Petra Cipher DB Agent, and Petra Cipher API Agent, uses the approved cryptographic algorithm of the validated cryptographic module KLIB V3.0 whose security and implementation conformance have been validated by Korea Cryptographic Module Validation Program (KCMVP).

1.4.2. Logical scope and boundary of the TOE

The logical scope of the TOE is as in [Figure 3] below.



[Figure 3] TOE Logical scope

1) Security Audit (FAU)

The Petra Cipher Key Server generates and manages audit data for security-relevant events in order to trace the responsibility for behaviors related to the security. The Audit data include the date and time of an event, the type of an event, subject identity, and event outcome(success of failure).

All generated audit data can only be accessed by authorized administrators. When viewing audit data, it is possible to selectively retrieve it by setting search criteria with logical relationships and by sorting. Among the audit data generated by the TOE, failures such as administrator authentication failures, integrity check failures, self-test failures, exceeding the predicted threshold for audit data repository capacity and saturation of the audit data repository capacity are detected as potential security violations. Then alerts are sent via email to the address specified by the authorized administrator.

If the audit data repository capacity exceeds the predicted threshold, an alert email is sent to the administrator. If the audit data repository capacity becomes saturated, the oldest audit data is overwritten. The audit records stored in the audit data repository are protected by preventing an unauthorized deletion of the stored audit data.

The Petra Cipher DB/API Agent generates audit data for audit events, including the date and time of an event, the type of an event, subject identity, and event outcome(success of failure). The generated audit data is sent to the Petra Cipher Key Server.

2) Cryptographic support (FCS)

The TOE securely generates and destroys all cryptographic keys used for product operation through a validated cryptographic module (KLIB V3.0), whose security and implementation conformance have been confirmed by the Korea Cryptographic Module Validation Program (KCMVP). It performs cryptographic operations based on the encryption policy that defines cryptographic algorithms. Additionally, to enable encrypted communication between physically separated TOE components, cryptographic keys are generated and exchanged via the validated cryptographic module.

When generating cryptographic keys, the TOE uses random numbers generated by the validated cryptographic module's random number generator (HASH_DRBG SHA-256). For encrypting user data in the protected DBMS, the TOE employs the algorithms listed in [Table 1-8] of the validated cryptographic module.

List of Standards	Cryptographic Operation Algorithm	Cryptographic Key Length	Use
KS X 1213-1	ARIA (CBC, OFB, CFB)	128	User data encryption/decryption (symmetric key encryption)
	ARIA (CBC, OFB)	192	
	ARIA (CBC, OFB)	256	
TTAS.KO-12.0004/R1 ISO/IEC 18033-3	SEED (CBC, OFB, CFB)	128	User data encryption/decryption (symmetric key encryption)
ISO/IEC 10118-3 ISO/IEC 10118-3 Amd 1	SHA-256	N/A	User data encryption (one-way encryption)
	SHA-384	N/A	
	SHA-512	N/A	

[Table 1-8] User data cryptographic operation algorithm

The TOE uses the algorithms listed in [Table 1-9] of the validated cryptographic module to protect TSF (TOE Security Function) data.

Cryptographic Operation Algorithm	Cryptographic Key Length (bit)	Use
ARIA (CBC)	256	Encryption of private key used for mutual authentication
		Encryption of private key used for encrypted communication
		Encryption of passwords of the authorized administrator
		Audit log encryption (Accessed administrator ID, accessed administrator IP, details)
		Encryption of TSF data cryptographic key
		Encryption of user data cryptographic key
RSA-PSS	2048	Encryption of digital signature for mutual authentication
SEED (CBC)	128	Encryption of the TOE internal communication data
SHA-256	N/A	Encryption of passwords of the authorized administrator
		TSF data encryption key Integrity check value
		User data encryption key Integrity check value
HMAC-SHA256	256	Generating an Encryption Key for the Master Key Using the Product Installation Password

[Table 1-9] TSF data cryptographic operation algorithm

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

3) User data protection (FDP)

When user data is stored in or retrieved from the protected DB, the Petra Cipher DB/API Agent encrypts/decrypts the user data by using the validated cryptographic module KLIB V3.0 according to the user data encryption/decryption policies established by the authorized administrator.

In case of the plug-in type, Petra Cipher DB Agent performs the user data encryption/decryption

at the column level. In case of the API type, Petra Cipher API Agent installed on the Application Server encrypts/decrypts user data.

The Petra Cipher DB/API Agent generates different encryption values for the same user data each time it performs the encryption.

When the encryption/decryption is complete, the Petra Cipher DB/API Agent carries out the initialization so that the previous original user data value cannot be recovered. (in case of data generated with a SHA algorithm, however, the algorithm itself does not support the decryption).

4) Identification and authentication (FIA)

The Petra Cipher Key Server provides the identification and authentication function for an administrator in charge of the security management. When data are entered to identify and authenticate the administrator, the password entered is masked with "●" to protect the authentication feedback. In addition, in case of failed authentication, feedback on the reason for failure is not provided. If authentication attempts fail consecutively (five times), the account is locked (for five minutes).

The Petra Cipher Key Server verifies that the administrator password meets defined criteria (length and composition rules) during creation or modification. The Petra Cipher Key Server also prevents the reuse of authentication data of the administrator logging in to the TOE.

The TOE performs mutual authentication through the protocol developed by SINSIWAY Co., Ltd. for the purpose of the secure communication among the TOE components(between the Petra Cipher Key Server and the Petra Cipher DB Agent, and between the Petra Cipher Key Server and the Petra Cipher API Agent).

5) Security Management (FMT)

The Petra Cipher Key Server has only one administrator account, changing the ID and password when the authorized administrator accesses for the first time.

The Petra Cipher Key Server provides the function of security management to manage user data encryption/decryption policies, cryptographic key, administrator information, audit record and agent configuration. The authorized administrator performs the security management through the security management interface.

6) Protection of the TSF (FPT)

The TOE performs encrypted communication using a validated cryptographic module(SEED-CBC 128 bit) to protect TSF data transmitted between physically separated TOE components(between the Petra Cipher Key Server and the Petra Cipher DB Agent, and between the Petra Cipher Key Server and the Petra Cipher API Agent).

The TOE encrypts and stores TSF data (e.g., encryption keys, passwords, configuration settings) using a validated cryptographic module (ARIA-CBC 256 bit, SHA-256) to protect it from exposure and modification.

To maintain the TOE's secure state and ensure that security functions operate correctly, it performs self-tests both at startup and periodically (every minute) during normal operation. Additionally, it provides authorized administrators with the capability to verify the integrity of TSF data and TSF executable code.

7) TOE access (FTA)

The Petra Cipher Key Server restricts the administrator's management access sessions whose access is allowed to perform the security management function to one. If a session of the administrator who has logged in to the Management Server already exists, the TOE can either terminate the existing session or block the new session based on the choice of the newly logged-in administrator.

If the authorized administrator logs in to the Petra Cipher Key Server via a web browser and then remains inactive for a period of time (10 minutes), the session that accessed the Petra Cipher Key Server is terminated.

The Petra Cipher Key Server allows management access only from administrator PCs configured with allowed IP addresses, and the number of permissible IP addresses is limited to two.

1.5. Terms and definitions

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

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

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

- Approved mode of operation
  An operation mode of a cryptographic module that uses an approved cryptographic algorithm

- Approved cryptographic algorithm
  A cryptographic algorithm selected by an institution that validates cryptographic modules taking into account the security, credibility, interoperability and so forth with regard to block cipher, hash function, message authentication code, random bit generator, key settings, public key encryption, and digital signature cryptographic algorithms

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

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

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

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

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

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

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

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

- Data Encryption Key (DEK)
  Key that encrypts and decrypts the data

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

- Security Function Policy (SFP)
  A set of rules that describes the specific security action performed by TSF (TOE security

functionality) and describe them as SFR (security function requirement)

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

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

- Security token
  Hardware device that implements key generation and digital signature generation inside the device to save/store confidential information safely

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

- Decryption
  The act that restores the ciphertext into the plaintext using the decryption key

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

- User
  Refer to "External entity"

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

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

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

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

- Element
  Indivisible statement of a security need

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

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

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

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

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

- Authorized administrator
  Authorized user who securely operates and manages the TOE

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

operation

- Authentication data
  Information used to verify the claimed identity of a user

- Self-test
  Pre-operational or conditional test executed by the cryptographic module

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

- Refinement
  Addition of details to a component

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

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

- Subject
  Active entity in the TOE that performs operations on objects

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

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

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

- Class
  Set of CC families that share a common focus

- Key Encryption Key (KEK)
  Key that encrypts and decrypts another cryptographic key

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

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

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

- Assignment
  Specification of an identified parameter in a component (of the CC) or requirement

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

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

- Critical Security Parameters (CSP)
  Information related to security that can erode the security of the cryptographic module if exposed or changed (e.g., verification data such as secret key/private key, password, or

Personal Identification Number)

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

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

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

- Secure Sockets Layer (SSL)
  This is a security protocol proposed by Netscape to ensure confidentiality, integrity and security over a computer network.

- Transport Layer Security (TLS)
  This is a cryptographic protocol between a SSL-based server and a client and is described in RFC 2246.

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

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

- Master Key
  It refers to the Key Encryption Key (KEK) used in Petra Cipher V3.2.

- SOHA Database
  DBMS developed by SINSIWAY Co., Ltd. and built inside Petra Cipher Key Server. SOHA Database is a memory DBMS that supports fast processing speeds and the safe security transaction processing.

1.6. Conventions

The operations used herein – selection, assignment, refinement and iteration – follow the same conventions specified in the CC.

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

Each operation is used in this ST.

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

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

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

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

2. Conformance Claim

2.1. CC, PP and Security requirement package conformance claim

This ST and the TOE confirm to the following CC, PP and the security requirement package:

Classification	Conformance
CC (Common Criteria)	Common Criteria for Information Technology Security Evaluation V3.1R5 ▪ Common Criteria Part 1: Introduction and General Model V3.1r5 (CCMB-2017-04-001, April 2017) ▪ Common Criteria Part 2: Security Functional Components V3.1r5 (CCMB-2017-04-002, April 2017) ▪ Common Criteria Part 3: Security Assurance Components V3.1r5 (CCMB-2017-04-003, April 2017)
CC Part 2 Security Functional Requirements	Extended: FCS_RBG.1, FIA_IMA.1, FDP_UDE.1, FMT_PWD.1, FPT_PST.1, FTA_SSL.5
CC Part 3 Security Assurance Requirements	Conformant
Security Requirement Package	Augmented: EAL1 augmented (ATE_FUN.1)
Protection Profile	National Protection Profile for Database Encryption V1.1 (December 2019)

2.2. Conformance claim rationale

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

Classification	PP	ST	Rationale
TOE Type	Database Encryption	Database Encryption	Same as the PP
Security Objective	OE.PHYSICAL_CONTROL	OE. PHYSICAL_CONTROL	Same as the security objectives for the operational environment in the PP
	OE.TRUSTED_ADMIN	OE.TRUSTED_ADMIN	Same as the security objectives for the operational environment in the PP
	OE.SECURE_DEVELOPMENT	OE.SECURE_DEVELOPMENT	Same as the security objectives for the operational environment in the PP

	OE.LOG_BACKUP	OE.LOG_BACKUP	Same as the security objectives for the operational environment in the PP
	OE.OPERATION_SYSTEM_ REINFORCEMENT	OE. OPERATION_SYSTEM_ REINFORCEMENT	Same as the security objectives for the operational environment in the PP
	-	OE.TIME_STAMP	Rationale described below the table
	-	OE.ADMINISTRATIVE_ ACCESS	Rationale described below the table
Security Functional Requirements	FAU_ARP.1	FAU_ARP.1	Same as the PP
	FAU_GEN.1	FAU_GEN.1	Same as the PP
	FAU_SAA.1	FAU_SAA.1	Same as the PP
	FAU_SAR.1	FAU_SAR.1	Same as the PP
	FAU_SAR.3	FAU_SAR.3	Same as the PP
	FAU_STG.1	FAU_STG.1	Same as the PP
	FAU_STG.3	FAU_STG.3	Same as the PP
	FAU_STG.4	FAU_STG.4	Same as the PP
	FCS_CKM.1(1)	FCS_CKM.1(1)	Same as the PP
	FCS_CKM.1(2)	FCS_CKM.1(2)	Same as the PP
	FCS_CKM.2	FCS_CKM.2	Same as the PP
	FCS_CKM.4	FCS_CKM.4	Same as the PP
	FCS_COP.1(1)	FCS_COP.1(1)	Same as the PP
	FCS_COP.1(2)	FCS_COP.1(2)	Same as the PP
	FCS_RBG.1(Extended)	FCS_RBG.1(Extended)	Same as the PP
	FDP_UDE.1(Extended)	FDP_UDE.1(Extended)	Same as the PP
	FDP_RIP.1	FDP_RIP.1	Same as the PP
	FIA_AFL.1	FIA_AFL.1	Same as the PP
	FIA_IMA.1(Extended)	FIA_IMA.1(Extended)	Same as the PP
	FIA_SOS.1	FIA_SOS.1	Same as the PP
	FIA_UAU.2	FIA_UAU.2	Rationale described below the table
	FIA_UAU.4	FIA_UAU.4	Same as the PP
	FIA_UAU.7	FIA_UAU.7	Same as the PP
	FIA_UID.2	FIA_UID.2	Rationale described below the table
	FMT_MOF.1	FMT_MOF.1	Same as the PP
	FMT_MTD.1	FMT_MTD.1	Same as the PP
	FMT_PWD.1(Extended)	FMT_PWD.1(Extended)	Same as the PP
	FMT_SMF.1	FMT_SMF.1	Same as the PP

	FMT_SMR.1	FMT_SMR.1	Same as the PP
	FPT_ITT.1	FPT_ITT.1	Same as the PP
	FPT_PST.1(Extended)	FPT_PST.1(Extended)	Same as the PP
	FPT_TST.1	FPT_TST.1	Same as the PP
	FTP_TRP.1	FTP_TRP.1	Same as the PP
	FTA_MCS.2	FTA_MCS.2	Same as the PP
	FTA_SSL.5(Extended)	FTA_SSL.5(Extended)	Same as the PP
	FTA_TSE.1	FTA_TSE.1	Same as the PP
Assurance Requirements	ASE_INT.1	ASE_INT.1	Same as the PP
	ASE_CCL.1	ASE_CCL.1	Same as the PP
	ASE_OBJ.1	ASE_OBJ.1	Same as the PP
	ASE_ECD.1	ASE_ECD.1	Same as the PP
	ASE_REQ.1	ASE_REQ.1	Same as the PP
	ASE_TSS.1	ASE_TSS.1	Same as the PP
	ADV_FSP.1	ADV_FSP.1	Same as the PP
	AGD_OPE.1	AGD_OPE.1	Same as the PP
	AGD_PRE.1	AGD_PRE.1	Same as the PP
	ALC_CMC.1	ALC_CMC.1	Same as the PP
	ALC_CMS.1	ALC_CMS.1	Same as the PP
	ATE_FUN.1	ATE_FUN.1	Same as the PP
	ATE_IND.1	ATE_IND.1	Same as the PP
	AVA_VAN.1	AVA_VAN.1	Same as the PP

[Table 2-1] Conformance rationale

※ OE.TIME_STAMP: Reliable time stamps from the operational environment are provided to accurately record security-relevant events. Therefore, its security objective is additionally defined.

※ OE.ADMINISTRATIVE_ACCESS: When an administrator attempts to connect to the Petra Cipher Key Server, which is a component of the TOE, all transmitted information must be securely protected. Therefore, an additional security objective is defined to ensure the protection of this communication.

※ The PP states that if no actions are appropriate in assignment operation of FIA_UAU.1.1, it is recommended to use FIA_UAU.2 which is in a hierarchical relationship with FIA_UAU.1. Therefore, FIA_UAU.2 has been used.

※ The PP states that if no actions are appropriate in assignment operation of FIA_UID.1.1, it is recommended to use FIA_UID.2 which is in a hierarchical relationship with FIA_UID.1. Therefore, FIA_UID.2 has been used.

3. Security Objectives

3.1. Security objectives for the operational environment

- OE.PHYSICAL_CONTROL

  The location where the TOE is installed and operated must be equipped with access control and protective equipment to ensure that only authorized administrators can access it. To maintain the independence of the system where Petra Cipher Key Server is installed, if it is installed on a dedicated security server, no other programs for different purposes should be installed on that server.

  - Secure Internal Network Management

    When installing Petra Cipher V4.0, firewalls must be configured on each server where the TOE is installed to ensure network protection. Unauthorized IPs should be blocked.

- OE.TRUSTED_ADMIN

  The authorized administrator of the TOE shall be non-malicious users, have been appropriately trained for the TOE management functions and accurately fulfill the duties in accordance with administrator guidance.

  - Blocking Unnecessary Services and User Accounts

    During the installation and operation of the TOE, care must be taken to avoid creating unnecessary services and user accounts (such as OS accounts, DB accounts, etc.). Any accounts that are not needed should be immediately deactivated or deleted to minimize system access. This practice helps reduce internal security threats and maintain system stability.

  - Secure File Permission Settings

    During TOE operation, appropriate file permissions must be set for sensitive data and system files. Each file's permissions should be set with the minimum necessary access to block unnecessary access. Any changes to file permissions must be logged to ensure traceability. In particular, critical configuration files and key files for encrypted communication must be accessible only to administrators.

  - Blocking Unnecessary Communication Ports

    During the installation and operation of the TOE, unused communication ports must be blocked to prevent unnecessary external access. Firewalls should be configured to open only the necessary ports, while unused ports should be blocked to enhance network security.

- OE.SECURE_DEVELOPMENT

  The developer who uses the TOE to interoperate with the encryption function in the application or the DBMS shall ensure that the security functions of the TOE are securely applied in accordance with the requirements of the manual provided with the TOE. .

- OE.LOG_BACKUP

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

- OE.OPERATION_SYSTEM_REINFORCEMENT

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

- OE.TIME_STAMP

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

- OE.MANAGEMENT_ACCESS

  All the data transmitted when an attempt is made to access Petra Cipher Key Server, which is a TOE component, shall be securely protected.

4. Extended Components Definition

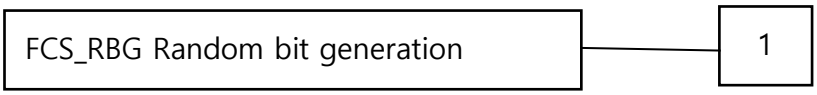
4.1. Cryptographic support (FCS)

4.1.1. Random bit generation

Family Behaviour

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

Component leveling



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

Management: FCS_RBG.1

There are no management activities foreseen.

Audit: FCS_RBG.1

There are no auditable events foreseen.

4.1.1.1. FCS_RBG.1 Random bit generation

Hierarchical to No other components.

Dependencies No dependencies.

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

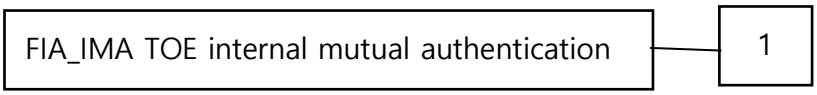
4.2. Identification & authentication (FIA)

4.2.1. TOE internal mutual authentication

Family Behaviour

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

Component Leveling



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

Management: FIA_IMA.1

There are no management activities foreseen.

Audit: FIA_IMA.1

The following actions are recommended to be recorded if FAU_GEN Security audit data generation:

a) Minimal: Success/failure of mutual authentication

b) Minimal: Modification of authentication protocol

4.2.1.1. FIA_IMA.1 TOE internal mutual authentication

Hierarchical to No other components.

Dependencies No dependencies.

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

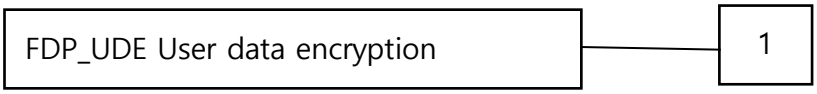
4.3. User data protection (FDP)

4.3.1. User data encryption

Family Behaviour

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

Component Leveling



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

Management: FDP_UDE.1

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

a) Management of user data encryption/decryption

Audit: FDP_UDE.1

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

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

4.3.1.1. FDP_UDE.1 User data encryption

Hierarchical to No other components.

Dependencies FCS_COP.1 Cryptographic operation

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

4.4. Security management (FMT)

4.4.1. ID and password

Family Behaviour

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

Component Leveling

FMT_PWD ID and password	1

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

Management: FMT_PWD.1

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

a) Management of ID and password configuration rules

Audit: FMT_PWD.1

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

a) Minimal: All changes of the passwords

4.4.1.1. FMT_PWD.1 Management of ID and password

Hierarchical to No other components.

Dependencies FMT_SMF.1 Specification of management functions

FMT_SMR.1 Security roles

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

1. [assignment: *password combination rules and/or length*]

2. [assignment: *other management such as management of special characters unusable for password, etc.*]

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

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

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

4.5. Protection of the TSF (FPT)

4.5.1. Protection of stored TSF data

Family Behaviour

This family (FPT_PST, Protection of Stored TSF data) defines rules to protect TSF data stored within containers controlled by the TSF from unauthorized modification or disclosure.

Component Leveling

FPT_PST Protection of stored TSF data	1

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

Management: FPT_PST.1

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

There are no management activities foreseen.

Audit: FPT_PST.1

There are no auditable events foreseen.

4.5.1.1. FPT_PST.1 Basic protection of stored TSF data

Hierarchical to No other components.

Dependencies No dependencies.

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

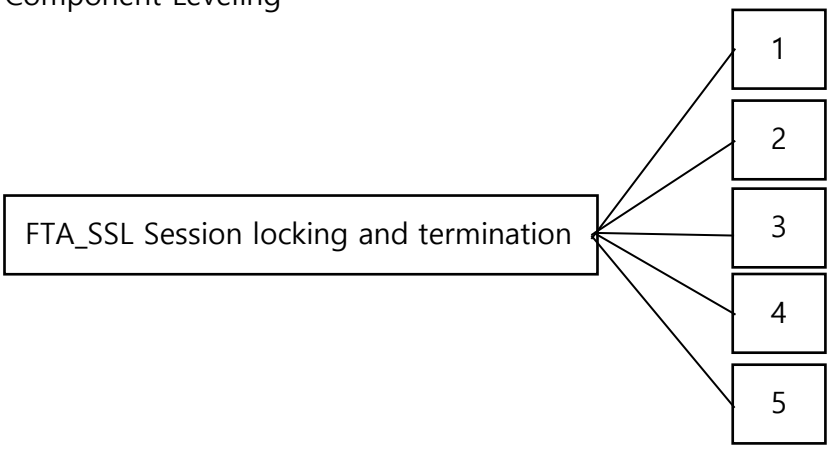
4.6. TOE access (FTA)

4.6.1. Session locking and termination

Family Behaviour

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

Component Leveling



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

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

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

Management: FTA_SSL.5

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

a) Specification for the time interval of user inactivity that results in session locking or termination for each user

b) Specification of the default user inactivity period that results in session locking or termination

Audit: FTA_SSL.5

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

a) Minimal: Locking or termination of interactive session

4.6.1.1. FTA_SSL.5 Management of TSF-initiated session

Hierarchical to No other components.

Dependencies [FIA_UAU.1 Authentication or no dependencies]

FTA_SSL.5.1 The TSF shall [selection:

- *lock the session and/or re-authenticate the,*
- *terminate*] an interactive session after [assignment: *time interval of user inactivity*].

5. Security Requirements

This chapter specifies the security functional requirements and the assurance requirements that must be satisfied by the TOE.

5.1. Security functional requirements

The security functional requirements defined in this ST are derived from relevant security functional components in CC Part 2 in order to satisfy the security objectives identified in Chapter 4. [Table 5-1] below summarizes the security functional components used in this ST.

Security Functional Class	Security Functional Component	
Security Audit (FAU)	FAU_ARP.1	Security alarms
	FAU_GEN.1	Audit data generation
	FAU_SAA.1	Potential violation analysis
	FAU_SAR.1	Audit review
	FAU_SAR.3	Selectable audit review
	FAU_STG.1	Protected audit trail storage
	FAU_STG.3	Action in case of possible audit data loss
	FAU_STG.4	Prevention of audit data loss
Cryptographic Support (FCS)	FCS_CKM.1(1)	Cryptographic key generation (User data encryption)
	FCS_CKM.1(2)	Cryptographic key generation (TSF data encryption)
	FCS_CKM.2	Cryptographic key distribution
	FCS_CKM.4	Cryptographic key destruction
	FCS_COP.1(1)	Cryptographic operation (User data encryption)
	FCS_COP.1(2)	Cryptographic operation (TSF data encryption)
	FCS_RBG.1(Extended)	Random bit generation
User Data Protection (FDP)	FDP_UDE.1(Extended)	User data encryption
	FDP_RIP.1	Subset residual information protection
Identification and Authentication (FIA)	FIA_AFL.1	Authentication failure handling
	FIA_IMA.1(Extended)	TOE internal mutual authentication
	FIA_SOS.1	Verification of secrets
	FIA_UAU.2	Authentication
	FIA_UAU.4	Single-use authentication mechanisms

	FIA_UAU.7	Protected authentication feedback
	FIA_UID.2	User identification before any action
Security Management (FMT)	FMT_MOF.1	Management of security functions behaviour
	FMT_MTD.1	Management of TSF data
	FMT_PWD.1(Extended)	Management of ID and password
	FMT_SMF.1	Specification of management functions
	FMT_SMR.1	Security roles
Protection of the TSF (FPT)	FPT_ITT.1	Basic internal TSF data transfer protection
	FPT_PST.1(Extended)	Basic protection of stored TSF data
	FPT_TST.1	TSF testing
TOE Access	FTA_MCS.2	Per user attribute limitation on multiple concurrent sessions
	FTA_SSL.5(Extended)	Management of TSF-initiated sessions
	FTA_TSE.1	TOE session establishment

[Table 5-1] Summary of security functional components

5.1.1. Security audit (FAU)

5.1.1.1. FAU_ARP.1 Security alarms

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

FAU_ARP.1.1 The TSF shall take [assignment: *an action to send an alarm message to the authorized administrator*] upon detection of a potential security violation.

5.1.1.2. FAU_GEN.1 Audit data generation

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

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

a) Start-up and shutdown of the audit function;
b) All auditable events for the *not specified* level of audit; and
c) ["Auditable events" in [Table 5-2], *none*].

FAU_GEN.1.2The TSF shall record within each audit record at least the following information:
a) Date and time of the event, type of event, subject identify (if applicable), and the outcome (success or failure) of the event, and
b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST [refer to the contents of "Additional Audit Record" in [Table 5-2], *no other components*].

Security Functional Component	Auditable Event	Additional Audit Record
FAU_ARP.1	Actions taken due to potential security violations	
FAU_SAA.1	Enabling and disabling of any of the analysis mechanisms, Automated responses performed by the tool	
FAU_STG.3	Actions taken due to exceeding of a threshold	
FAU_STG.4	Actions taken due to the audit storage failure	
FCS_CKM.1(1)	Success and failure of the activity	
FCS_CKM.2	Success and failure of the activity (applied only to distribution of key related to encryption/decryption of user data)	
FCS_CKM.4	Success and failure of the activity (applied only to the destruction of key related to encryption/decryption of user data)	
FCS_COP.1(1)	Success and failure of cryptographic operation, type of cryptographic operation	
FDP_UDE.1(Extended)	Success and failure of encryption/decryption of user data	
FIA_AFL.1	The reaching of the threshold for the unsuccessful authentication attempts and the actions taken and the subsequent, if appropriate, restoration to the normal state	
FIA_IMA.1(Extended)	Success/failure of mutual authentication	
FIA_UAU.2	All use of authentication mechanism	

FIA_UAU.4	Attempts to reuse authentication data	
FIA_UID.2	All use of the user identification mechanism, including the user identity provided	
FMT_MOF.1	All modifications in the behavior of the functions in the TSF	
FMT_MTD.1	All modifications to the values of TSF data	Modified TSF data value
FMT_PWD.1(Extended)	All changes of the password	
FMT_SMF.1	Use of the management functions	
FPT_TST.1	Execution of the TSF self-tests and the results of the tests	Modified TSF data or executable code in case of integrity violation
FTA_MCS.2	Denial of a new session based on the limitation of multiple concurrent sessions	
FTA_SSL.5(Extended)	Termination of interactive sessions	
FTA_TSE.1	Denial of session establishment due to the session mechanism, All attempts to establish a user session	

[Table 5-2] Auditable events

5.1.1.3. FAU_SAA.1 Potential violation analysis

Hierarchical to No other components.

Dependencies FAU_GEN.1 Audit data generation

FAU_SAA.1.1 The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the enforcement of the SFRs.

FAU_SAA.1.2 The TSF shall enforce the following rules for monitoring audited events: a) Accumulation or combination of [authentication failure audit event among auditable events of **FIA_UAU.2**, integrity violation audit event self-test failure event of the validated cryptographic module among auditable events of FPT_TST.1, [audit storage capacity exceeding the predefined threshold among auditable events in FAU_STG.3, overwriting of the oldest audit record if audit trail is full among auditable events in

FAU_STG.4] known to indicate a potential violation;
b) [None]

5.1.1.4. 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.

5.1.1.5. FAU_SAR.3 Selectable audit review

Hierarchical to No other components.
Dependencies FAU_SAR.1 Audit review

FAU_SAR.3.1 The TSF shall provide the capability to apply [methods of ordering specified in [Table 5-4]] of audit data based on [criteria with logical relations specified in [Table 5-3]].

Audit Data Type	Audit Data Column Name	Criteria for Logical Relations
Administrator task history	Date and time of the task	Range search
	Category, Menu, Function, Content, ID, IP	AND search
Server history	Date and time of the task	Range search
	Content, Log	AND search
Encryption Key Request history	Date and time of the task	Range search
	Encrypt Key Id, Encrypt Key name, Algorithm, Encrypt mode, Request IP	AND search
Encryption task history	Date and time of the task	Range search
	DB, DB User, Schema, Table, Column, user IP, access program	AND search

[Table 5-3] Criteria for selection by audit data type

Audit Data Type	Audit Data Column Name	Method of Ordering
Administrator task history	Date and time of the task	Default descending sorting It changes to an ascending order if clicking the column. It is sorted out on one column per type.
	Category, Menu, Function, Content, ID, IP	No default sorting If changes to an ascending order if clicking the column, and changes to a descending order upon the next click.
Server history	Date and time of the task	Default descending sorting It changes to an ascending order if clicking the column. It is sorted out on one column per type.
	Content, Log	No default sorting If changes to an ascending order if clicking the column, and changes to a descending order upon the next click. It is sorted out on one column per type.
Encryption Key Request history	Date and time of the task	Default descending sorting It changes to an ascending order if clicking the column. It is sorted out on one column per type.
	Encrypt Key Id, Encrypt Key name, Algorithm, Encrypt mode, Request IP	No default sorting If changes to an ascending order if clicking the column, and changes to a descending order upon the next click. It is sorted out on one column per type.
Encryption task history	Date and time of the task	Default descending sorting It changes to an ascending order if clicking the column. It is sorted out on one column per type.
	DB, DB User, Schema, Table, Column, user	No default sorting

	IP, access program	If changes to an ascending order if clicking the column, and changes to a descending order upon the next click. It is sorted out on one column per type.

[Table 5-4] Method or ordering per audit data type

5.1.1.6. FAU_STG.1 Protected audit trail storage

Hierarchical to No other components.

Dependencies FAU_GEN.1 Audit data generation

FAU_STG.1.1 The TSF shall protect the stored audit data in the audit trail from unauthorized deletion.

FAU_STG.1.2 The TSF shall be able to *prevent* unauthorized modifications to the stored audit records in the audit trail.

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

Hierarchical to No other components.

Dependencies FAU_STG.1 Protected audit data trail storage

FAU_STG.3.1 The TSF shall [notify the authorized administrator, [none]] if the audit trail exceeds [a threshold value defined by the administrator (the threshold can be an integer number between 50 and 80, in the unit of percent (%)), default value: 80% of the audit storage capacity].

5.1.1.8. FAU_STG.4 Prevention of audit data loss

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

Dependencies FAU_STG.1 Protected audit trail storage

FAU_STG.4.1 The TSF shall [*overwrite the oldest stored audit records* and *send an alarm email to the authorized administrator*] if the audit trail is full.

5.1.2. Cryptographic support (FCS)

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

Hierarchical to Dependencies: No other components.
[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [cryptographic key generation algorithm in [Table 5-5]] and specified cryptographic key sizes [cryptographic key sizes in [Table 5-6]] that meet the following [list of standards in [Table 5-5]].

List of Standards	Cryptographic Key Generation Algorithm	Cryptographic Key Size	Use
NIST SP 800-90	HASH_DRBG SHA-256	N/A	Encryption key generation algorithm

[Table 5-5] User data cryptographic key generation algorithm and list of standards

List of Standards	Cryptographic Operation Algorithm	Cryptographic Key Size	Use
KS X 1213-1	ARIA (CBC, OFB, CFB)	128	User data encryption/decryption (asymmetric key encryption)
	ARIA (CBC, OFB)	192	
	ARIA (CBC, OFB)	256	
TTAS.KO-12.0004/R1 ISO/IEC 18033-3	SEED (CBC, OFB, CFB)	128	User data encryption/decryption (symmetric key encryption)

[Table 5-6] User data cryptographic key operation algorithm and list of standards

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

Hierarchical to: No other components.

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

FCS_CKM.1.1The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [cryptographic key generation algorithm in [Table 5-7]] and specified cryptographic key sizes [cryptographic key sizes in [Table 5-8]] that meet the following [list of standards in [Table 5-7]].

List of Standards	Cryptographic Key Generation Algorithm	Cryptographic Key Size	Use
NIST SP 800-90	HASH_DRBG SHA-256	N/A	Cryptographic key generation algorithm

[Table 5-7] TSF data cryptographic key generation algorithm and list of standards

List of Standards	Cryptographic Key Operation Algorithm	Cryptographic Key Size	Use
KS X 1213-1	ARIA (CBC)	256	Encryption of private key used for mutual authentication
			Encryption of private key used for encrypted communication
			Encryption of password of authorized administrator
			Encryption of audit log (accessed administrator ID, accessed administrator IP, details)
			TSF data cryptographic key encryption
			User data cryptographic key encryption
ISO/IEC 14888-2	RSA-PSS	2048	Encryption of mutual authentication digital signature
TTAS.KO-12.0004/R1	SEED (CBC)	128	Encryption of the TOE internal

ISO/IEC 18033-3			communication data
ISO/IEC 9797-2	HMAC-SHA256	1024	Generating an Encryption Key for the Master Key Using the Product Installation Password

[Table 5-8] TSF data cryptographic operation algorithm and list of standards

5.1.2.3. FCS_CKM.2 Cryptographic key distribution

Hierarchical to No other components.

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

FCS_CKM.2.1The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method [cryptographic key distribution algorithm in Table [5-9]] that meets the following [list of standards in [Table 5-9]].

List of Standards	Cryptographic Key Distribution Algorithm	Cryptographic Key Size	Use
ISO/IEC 18033-2	RSAES	2048	Cryptographic key distribution

[Table 5-9] Cryptographic key distribution algorithm and list of standards

5.1.2.4. FCS_CKM.4 Cryptographic key destruction

Hierarchical to No other components.

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

FCS_CKM.4.1The TSF shall destroy cryptographic key in accordance with a specified cryptographic key destruction method [overwriting with 0] that meets the following [none].

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

Hierarchical to No other components.

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

FCS_COP.1.1 The TSF shall perform [use in [Table 5-10]] in accordance with a specified cryptographic algorithm [cryptographic operation algorithm in [Table 5-10]] and cryptographic key size [cryptographic key size in [Table 5-10]] that meet the following [list of standards in [Table 5-10]].

List of Standards	Cryptographic Operation Algorithm	Cryptographic Key Size	Use
KS X 1213-1	ARIA (CBC, OFB, CFB)	128	User data encryption/decryption (symmetric key encryption)
	ARIA (CBC, OFB)	192	
	ARIA (CBC, OFB)	256	
TTAS.KO-12.0004/R1 ISO/IEC 18033-3	SEED (CBC, OFB, CFB)	128	User data encryption/decryption (symmetric key encryption)
ISO/IEC 10118-3 ISO/IEC 10118-3 Amd 1	SHA-256	N/A	User data encryption (one-way encryption)
	SHA-384	N/A	
	SHA-512	N/A	

[Table 5-10] User data cryptographic operation algorithm and list of standards

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

Hierarchical to No other components.

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

FCS_COP.1.1 The TSF shall perform [use in [Table 5-11]] in accordance with a specified

cryptographic algorithm [cryptographic operation algorithm in [Table 5-11]] and cryptographic key size [cryptographic key size in [Table 5-11]] that meet the following [list of standards in [Table 5-11]].

List of Standards	Cryptographic Operation Algorithm	Cryptographic Key Size	Use
KS X 1213-1	ARIA (CBC)	256	Encryption of private key used for mutual authentication
			Encryption of private key used for encrypted communication
			Encryption of password of authorized administrator
			Encryption of audit log (accessed administrator ID, accessed administrator IP, details)
			TSF data cryptographic key encryption
			User data cryptographic key encryption
ISO/IEC 14888-2	RSA-PSS	2048	Encryption of mutual authentication digital signature
TTAS.KO-12.0004/R1 ISO/IEC 18033-3	SEED (CBC)	128	Encryption of the TOE internal communication data
ISO/IEC 10118-3 ISO/IEC 10118-3 Amd 1	SHA-256	N/A	Encryption of password of authorized administrator
			TSF data encryption key Integrity check value
			User data encryption key Integrity check value
ISO/IEC 9797-2	HMAC-SHA256	256	Generating an Encryption Key for the Master Key Using the Product Installation Password

[Table 5-11] TSF data cryptographic operation algorithm and list of standards

5.1.2.7. FCS_RBG.1 Random bit generation (Extended)

Hierarchical to No other components.

Dependencies No dependencies.

FCS_RBG.1.1 The TSF shall generate random bits using the specified random bit generator that meets the following [NIST SP 800-90].

5.1.3. User data protection (FDP)

5.1.3.1. FDP_UDE.1 User data encryption

Hierarchical to No other components.

Dependencies FCS_COP.1 Cryptographic operation

FDP_UDE.1.1 The TSF shall provide a function that can encrypt/decrypt the user data to the TOE user according to the specified [column-level encryption/decryption method, [none]].

5.1.3.2. FDP_RIP.1 Subset residual information protection

Hierarchical to No other components.

Dependencies No dependencies.

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

5.1.4. Identification and Authentication (FIA)

5.1.4.1. FIA_AFL.1 Authentication failure handling

Hierarchical to No other components.

Dependencies FIA_UAU.1 Timing of authentication

FIA_AFL.1.1 The TSF shall detect when *[five]* unsuccessful authentication attempts occur related to [administrator authentication attempt].

FIA_AFL.1.2 When the defined number of authentication attempts has been *met*, the TSF shall [inactivate the identification and authentication function for five minutes].

5.1.4.2. 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 using [authentication protocol implemented by SINSIWAY Co., Ltd.] in accordance with [none] between [authentication between TOE components in [Table 5-12]].

Mutual authentication between TOE components
Petra Cipher Key Server <-> Petra Cipher DB Agent
Petra Cipher Key Server <-> Petra Cipher API Agent

[Table 5-12] Mutual authentication between TOE components

5.1.4.3. FIA_SOS.1 Verification of secrets

Hierarchical to No other components.

Dependencies No dependencies.

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet [a defined quality metric in [Table 13]].

Classification		Defined Quality Metric
Password combination rules	General combination rule	Password has a combination of three types of characters - alphabets, numbers and special characters. The length shall be at least 9 digits up to 13 digits.
	Number (10 numbers)	0-9
	Character (52 alphabets)	English upper case (26 alphabets) A-Z, English lower case (26 alphabets) a-z
	Special Characters (22 letters)	~ ! @ # $ % ^ & * _ + ` - { } : < > ? , . /

[Table 5-13] Password combination rule

5.1.4.4. FIA_UAU.2 User authentication before any action

Hierarchical to FIA_UAU.1 Timing of authentication

Dependencies FIA_UID.1 Timing of identification

FIA_UAU.2.1 The TSF shall require the **authorized administrator** to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

5.1.4.5. FIA_UAU.4 Single-use authentication mechanisms

Hierarchical to No other components.

Dependencies No dependencies.

FIA_UAU.4.1 The TSF shall prevent reuse of authentication data related to [password authentication mechanism].

5.1.4.6. 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 [feedback to mask password being entered ("●") and provide an error message that "Please check your ID or Password." in case of failed authentication] to the user while the authentication is in progress.

5.1.4.7. FIA_UID.2 User identification before any action

Hierarchical to FIA_UID.1 Timing of identification

Dependencies No dependencies.

FIA_UID.2.1 The TSF shall require each **authorized administrator** to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

5.1.5. Security management (FMT)

5.1.5.1. FMT_MOF.1 Management of security functions behavior

Hierarchical to No other components.

Dependencies FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

FMT_MOF.1.1 The TSF shall restrict the ability to ***conduct management actions of*** the functions [list of security functions of the administrator in [Table 5-14]] to the [authorized administrator].

Security Functional Component	Management Function	Management Type
FAU_SAR.1, FAU_SAR.3	Function to view administrator task history, server history and encryption task history	Management of security functions
FCS_CKM.1(1), FCS_CKM.4	Function to add/delete/modify an encryption key used for user data encryption/decryption	Management of security attributes
FDP_UDE.1	Management of rules for user data encryption/decryption block	Management of security attributes
FMT_PWD.1	Function to change password of the authorized administrator	Management of security functions
FDP_UDE.1	Function to add/delete/modify information of the protected database when performing plug-in encryption	Management of security functions
FDP_UDE.1	Function to collect table and column information of the protected database when performing plug-in encryption	Management of security functions
FDP_UDE.1	Function to designate a column to be encrypted in the protected database when performing plug-in encryption	Management of security functions
FDP_UDE.1	Function to encrypt a column to be encrypted when performing plug-in encryption	Management of security

		functions
FDP_UDE.1	Function to register the library agent (Petra Cipher DB Agent, Petra Cipher API Agent) of the TOE and check the connection state when performing the encryption	Management of security functions
FPT_TST.1	Function to verify the integrity	Management of security functions

[Table 5-14] List of security function behavior of the administrator

5.1.5.2. FMT_MTD.1 Management of TSF data

Hierarchical to No other components.

Dependencies FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

FMT_MTD.1.1 The TSF shall restrict the ability to *manage* [[Table 5-15] List of TSF data management behaviors of the administrator] to the [authorized administrator].

Security Functional Component	Management Function	Management Type
FAU_SAA.1	Modify information on the mail server that sends alarm mails, sender mails and receiver mails	Management of TSF data
FAU_STG.3	Modify the threshold value of the function to notify the threshold of the audit trail storage disk	Management of TSF data threshold
FAU_STG.4	Modify the threshold value of the overwriting if the audit trail storage disk is full	
FAU_STG.3, FAU_STG.4	Modify the interval to check the capacity of the audit trail storage disk	Management of TSF data
FTA_TSE.1	Add/modify allowed IP of the administrator PC which an authorized user can access	Management of TSF data

[Table 5-15] List of TSF data management behaviors of the administrator

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

Hierarchical to No other components.

Dependencies FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

FMT_PWD.1.1 The TSF shall restrict the ability to manage the password of [none] to [none].
1. [None]
2. [None]

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

FMT_PWD.1.3 The TSF shall provide the capability for *changing the ID and password when the authorized administrator accesses for the first time*.

5.1.5.4. FMT_SMF.1 Specification of management functions

Hierarchical to No other components.

Dependencies No dependencies.

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: [
Management of security functions: Management functions specified in FMT_MOF.1,
Management of TSF data: Management functions specified in FMT_MTD.1,
Management of ID and password: Management functions specified in FMT_PWD.1]

5.1.5.5. 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 [authorized administrator].

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

5.1.6. Protection of the TSF (FPT)

5.1.6.1. FPT_ITT.1 Basic internal TSF data transfer protection

Hierarchical to No other components.

Dependencies No dependencies.

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

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

Hierarchical to No other components.

Dependencies No dependencies.

FPT_PST.1.1 The TSF shall protect [TSF data:
- Audit log,
- Administrator authentication data,
- Encryption key information (master key, private key, symmetric key),
- TOE setting value

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

5.1.6.3. FPT_TST.1 TSF testing

Hierarchical to No other components.

Dependencies No dependencies.

FPT_TST.1.1 The TSF shall run a suite of self-tests *during initial start-up, periodically during normal operation* to demonstrate the correct operation of *[TOE self-tests items in [Table 5-16]].*

FPT_TST.1.2 The TSF shall provide the **authorized administrator** with the capability

to verify the integrity of *[configuration file in [Table 5-17] TOE integrity test items]*.

FPT_TST.1.3 The TSF shall provide the **authorized administrator** with the capability to verify the integrity of *[executable file in [Table 5-17] TOE integrity test items]*.

TOE Classification	Self-test Item	Test Description
Petra Cipher Key Server	Validated cryptographic module	Self-test inside the validated cryptographic module
	Process	Check whether main processes necessary for the operation of Petra Cipher Key Server are in normal operation, and then send the test result to the audit log
Petra Cipher DB Agent	Validated cryptographic module	Self-test inside the validated cryptographic module
	Process	Check whether main processes necessary for the operation of Petra Cipher DB Agent are in normal operation, and then send the test result to the audit log
Petra Cipher API Agent	Validated cryptographic module	Self-test inside the validated cryptographic module
	Process	Check whether main processes necessary for the operation of Petra Cipher API Agent are in normal operation, and then send the test result to the audit log

[Table 5-16] TOE self-test items

TOE Classification	Integrity Test Item	Test Description
Petra Cipher Key Server	Executable file, configuration file	Check whether the executable file of Petra Cipher Key Server was corrupted by an unauthorized user, and the send the test result to the audit log
Petra Cipher DB Agent	Executable file, configuration file	Check whether the executable file of Petra Cipher DB Agent was corrupted by an unauthorized user, and the send the test result to the audit log

Petra Cipher API Agent	Executable file, configuration file	Check whether the executable file of Petra Cipher API Agent was corrupted by an unauthorized user, and the send the test result to the audit log

[Table 5-17] TOE integrity test items

5.1.7. TOE access (FTA)

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

Hierarchical to FTA_MCS.1 Basic limitation on multiple concurrent sessions

Dependencies FIA_UID.1 Timing of identification

FTA_MCS.2.1 The TSF shall restrict the maximum number of concurrent sessions [belonging to the same **administrator** according to the rules for the list of management functions defined in FMT_SMF.1.1]

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

b) Limit the maximum number of concurrent sessions to {0} for management access by the same administrator who doesn't have the right to perform FMT_MOF.1.1 "Management actions" but has the right to perform a query in FMT_MTD.1.1 "Management" only.

c) [none]

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

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

Hierarchical to No other components.

Dependencies FIA_UAU.1 Authentication or No dependencies.

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

5.1.7.3. FTA_TSE.1 TOE session establishment

Hierarchical to No other components.

Dependencies No dependencies.

FTA_TSE.1.1 The TSF shall be able to refuse the **management access session of the administrator**, based on [access IP, [*none*]].

5.2. Security assurance requirements

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

Security Assurance Class	Security Assurance Component	
Security Target Evaluation	ASE_INT.1	ST introduction
	ASE_CCL.1	Conformance claims
	ASE_OBJ.1	Security objectives for the operational environment
	ASE_ECD.1	Extended components definition
	ASE_REQ.1	Stated security requirements
	ASE_TSS.1	TOE summary specification
Development	ADV_FSP.1	Basic functional specification
Guidance Document	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative procedures
Life-cycle Support	ALC_CMC.1	Labeling of the TOE
	ALC_CMS.1	TOE configuration management coverage
Tests	ATE_FUN.1	Functional testing
	ATE_IND.1	Independent testing: conformance
Vulnerability Assessment	AVA_VAN.1	Vulnerability survey

[Table 5-18] Assurance component summary

5.2.1. Security Target evaluation

5.2.1.1. ASE_INT.1 ST introduction

Dependencies No dependencies.

Developer action elements

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

Content and presentation elements

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

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

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

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

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

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

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

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

Evaluator action elements

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

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

5.2.1.2. ASE_CCL.1 Conformance claims

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

Developer action elements

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

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

Content and presentation elements

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

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

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

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

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

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

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

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

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

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

Evaluator action elements

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

5.2.1.3. ASE_OBJ.1 Security objectives for the operational environment

Dependencies No dependencies.

Developer action elements

ASE_OBJ.1.1D The developer shall provide a statement of security objectives.

Content and presentation elements

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

Evaluator action elements

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

5.2.1.4. ASE_ECD.1 Extended components definition

Dependencies No dependencies.

Developer action elements

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

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

Content and presentation elements

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

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

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

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

Evaluator action elements

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

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

5.2.1.5. ASE_REQ.1 Stated security requirements

Dependencies ASE_ECD.1 Extended components definition

Developer action elements

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

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

Content and presentation elements

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

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

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

ASE_REQ.1.4C All operations shall be performed correctly.

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

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

Evaluator action elements

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

5.2.1.6. ASE_TSS.1 TOE summary specification

Dependencies ASE_INT.1 ST introduction

ASE_REQ.1 Stated security requirements

ADV_FSP.1 Basic functional specification

Developer action elements

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

Content and presentation elements

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

Evaluator action elements

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

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

5.2.2. Development

5.2.2.1. ADV_FSP.1 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 categorisation of interfaces as SFR-non-interfering.

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

Evaluator action elements

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

ADV_FSP.1.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

5.2.3. Guidance documents

5.2.3.1. AGD_OPE.1 Operational user guidance

Dependencies ADV_FSP.1 Basic functional specification

Developer action elements

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

Content and presentation elements

AGD_OPE.1.1C The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure

processing environment, including appropriate warnings.

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

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

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

AGD_OPE.1.5C The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

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

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

Evaluator action elements

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

5.2.3.2. AGD_PRE.1 Preparative procedures

Dependencies No dependencies.

Developer action elements

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

Content and presentation elements

AGD_PRE1.1C The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE1.2C The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational

environment in accordance with the security objectives for the operational environment as described in the ST.

Evaluator action elements

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

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

5.2.4. Life-cycle support

5.2.4.1. ALC_CMC.1 Labeling of the TOE

Dependencies ALC_CMS.1 TOE CM coverage

Developer action elements

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

Content and presentation elements

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

Evaluator action elements

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

5.2.4.2. ALC_CMS.1 TOE CM coverage

Dependencies No dependencies.

Developer action elements

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

Content and presentation elements

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

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

Evaluator action elements

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

5.2.5. Tests

5.2.5.1. ATE_FUN.1 Functional testing

Dependencies ATE_COV.1 Evidence of coverage

Developer action elements

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

ATE_FUN.1.2D The developer shall provide test documentation.

Content and presentation elements

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

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

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

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

Evaluator action elements

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

5.2.5.2. ATE_IND.1 Independent testing: conformance

Dependencies ADV_FSP.1 Basic functional specification

AGD_OPE.1 Operational user guidance

AGD_PRE.1 Preparative procedures

Developer action elements

ATE_IND.1.1D The developer shall provide the TOE for testing.

Content and presentation elements

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

Evaluator action elements

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

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

5.2.6. Vulnerability assessment

5.2.6.1. AVA_VAN.1 Vulnerability survey

Dependencies ADV_FSP.1 Basic functional specification

AGD_OPE.1 Operational user guidance

AGD_PRE.1 Preparative procedures

Developer action elements

AVA_VAN.1.1D The developer shall provide the TOE for testing.

Content and presentation elements

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

Evaluator action elements

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

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

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

5.3. Dependencies of the SFRs

The following [Table 5-19] shows dependencies of security functional requirements.

No.	SFR	Dependencies	Reference No.
1	FAU_ARP.1	FAU_SAA.1	3
2	FAU_GEN.1	FPT.STM.1	OE.TIME_STAMP
3	FAU_SAA.1	FAU_GEN.1	2
4	FAU_SAR.1	FAU_GEN.1	2
5	FAU_SAR.3	FAU_SAR.1	4
6	FAU_STG.1	FAU_GEN.1	2
7	FAU_STG.3	FAU_STG.1	6
8	FAU_STG.4	FAU_STG.1	6
9	FCS_CKM.1(1)	[FCS_CKM.2 or FCS_COP.1]	11, 13
		FCS_CKM.4	12
10	FCS_CKM.1(2)	[FCS_CKM.2 or FCS_COP.1]	11, 14
		FCS_CKM.4	12
11	FCS_CKM.2	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	9, 10
		FCS_CKM.4	12
12	FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	9, 10
13	FCS_COP.1(1)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	9
		FCS_CKM.4	12
14	FCS_COP.1(2)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	10
		FCS_CKM.4	12
15	FCS_RBG.1	-	-
16	FDP_UDE.1	FCS_COP.1	13
17	FDP_RIP.1	-	-
18	FIA_AFL.1	FIA_UAU.1	21
19	FIA_IMA.1	-	-
20	FIA_SOS.1	-	-
21	FIA_UAU.2	FIA_UID.1	24
22	FIA_UAU.4	-	-
23	FIA_UAU.7	FIA_UAU.1	21
24	FIA_UID.2	-	-
25	FMT_MOF.1	FMT_SMF.1	28
		FMT_SMR.1	29
26	FMT_MTD.1	FMT_SMF.1	28
		FMT_SMR.1	29
27	FMT_PWD.1	FMT_SMF.1	28
		FMT_SMR.1	29

28	FMT_SMF.1	-	-
29	FMT_SMR.1	FIA_UID.1	24
30	FPT_ITT.1	-	-
31	FPT_PST.1	-	-
32	FPT_TST.1	-	-
33	FTA_MCS.2	FIA_UID.1	24
34	FTA_SSL.5	FIA_UAU.1	21
35	FTA_TSE.1	-	-

[Table 5-19] Rationale for dependencies of the SFRs of the TOE

- FAU_GEN.1 has a dependency on FPT_STM.1. However, the TOE uses reliable time stamps provided in the TOE operational environment and accurately records audit data related to the operation of the TOE. Thus, the dependency of FAU_GEN.1 is satisfied by OE. TIME_STAMP, which is the security objective for the operational environment, on behalf of FPT_STM.1.
- FIA_AFL.1 has a dependency on FIA_UAU.1, which is satisfied by FIA_UAU.2 hierarchical to FIA_UAU.1.
- FIA_UAU.2 has a dependency on FIA_UID.1, which is satisfied by FIA_UID.2 hierarchical to FIA_UID.1.
- FIA_UAU.7 has a dependency on FIA_UAU.1, which is satisfied by FIA_UAU.2 hierarchical to FIA_UAU.1.
- FMT_SMR.1 has a dependency on FIA_UID.1, which is satisfied by FIA_UID.2 hierarchical to FIA_UID.1.
- FTA_MCS.2 has a dependency on FIA_UID.1, which is satisfied by FIA_UID.2 hierarchical to FIA_UID.1.
- FTA_SSL.5 has a dependency on FIA_UAU.1, which is satisfied by FIA_UAU.2 hierarchical to FIA_UAU.1.

5.4. Dependency of SFRs

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

The augmented SAR ATE_FUN.1 has a dependency on ATE_COV.1. ATE_FUN.1 has been augmented to ensure that the developer performs tests on test items correctly and documents them in the test documentation. However, ATE_COV.1 is not included in this ST since it is deemed not necessarily required to include ATE_COV.1 that presents the consistency between test items and TSFI.

6. TOE Summary Specification

This chapter summarizes security functionality required by the TOE.

6.1. Security audit

The TOE generates, records, and reviews audit records of security-relevant events in order to trace the accountability of behaviors related to the security. Furthermore, it detects potential security violations related to the audited events and takes actions in response. If the audit trail exceeds the threshold or if the audit trail is full, the TOE takes actions in a pre-defined manner.

6.1.1. Audit data generation

Petra Cipher Key Server, Petra Cipher DB Agent, and Petra Cipher API Agent, which are the TOE components, generate audit data. The audit data is transmitted to Petra Cipher Key Server, and stored in SOHA (DBMS developed by SINSIWAY Co., Ltd.) storage inside Petra Cipher Key Server.

Types of the generated audit data are listed in [Table 6-1] auditable events. Each audit data outputs the date and time of the event, type, the subject identity, and an event outcome.

No.	SFR	Auditable Event
1	FAU_ARP.1	Actions taken due to potential security violations
2	FAU_SAA.1	Enabling and disabling of any of the analysis mechanisms, Automated responses performed by the tool
3	FAU_STG.3	Actions taken due to exceeding of a threshold
4	FAU_STG.4	Actions taken due to the audit storage failure
5	FCS_CKM.1(1)	Success and failure of the activity
6	FCS_CKM.2	Success and failure of the activity (applied only to distribution of key related to encryption/decryption of user data)
7	FCS_CKM.4	Success and failure of the activity (applied only to the destruction of key related to encryption/decryption of user data)
8	FCS_COP.1(1)	Success and failure of cryptographic operation, type of cryptographic operation
9	FDP_UDE.1	Success and failure of encryption/decryption of user data
10	FIA_AFL.1	The reaching of the threshold for the unsuccessful authentication attempts

		and the actions taken and the subsequent, if appropriate, restoration to the normal state
11	FIA_IMA.1	Success/failure of mutual authentication, modification of authentication protocol
12	FIA_UAU.2	All use of authentication mechanism
13	FIA_UAU.4	Attempts to reuse authentication data
14	FIA_UID.2	All use of the user identification mechanism, including the user identity provided
15	FMT_MOF.1	All modifications in the behavior of the functions in the TSF
16	FMT_MTD.1	All modifications to the values of TSF data (modified values of TSF data)
17	FMT_PWD.1	All changes of the password
18	FMT_SMF.1	Use of the management functions
19	FPT_TST.1	Execution of the TSF self-tests and the results of the tests (modified TSF data or execution code in case of integrity violation)
20	FTA_MCS.2	Denial of a new session based on the limitation of multiple concurrent sessions
21	FTA_SSL.5	Locking or termination of interactive session
22	FTA_TES.1	Session establishment denial due to the session mechanism, All attempts to establish a user session

[Table 6-1] List of audit data generation

SFR to be satisfied: FAU_GEN.1

6.1.2. Potential violation analysis and action

The TOE sends an alarm mail to the email set by the authorized administrator in case of a potential security violation defined in FAU_SSA.1.

1) Authentication failure audit event among auditable events of FIA_UAU.2
2) Integrity violation audit event self-test failure event of the validated cryptographic module among auditable events of FPT_TST.1
3) Audit storage capacity exceeding the predefined threshold among auditable events in FAU_STG.3
4) Overwriting of the oldest audit record if audit trail is full among auditable events in FAU_STG.4

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

6.1.3. Management of audit storage

The TOE uses SOHA Database developed by SINSIWAY Co., Ltd. in order to protect the audit trail storage, and protects the stored audit data by blocking access to the database by an unauthorized user.

The TOE uses the entire available capacity of the disk partition in which the TOE is installed when piling up the audit data.

The TOE checks the disk partition in which the TOE is installed according to the defined interval of checking the audit data storage (default value: 60 seconds). If the disk space usage exceeds the predefined threshold, an alarm is sent to the mail server and the receiver email set by the authorized administrator.

- Alarm-triggering threshold: It can be set with an integer value between 50 and 80, in the unit of percent (%). The default value before setting is 50%.

If the audit trail is full (which means it reaches the threshold of overwriting predefined by the authorized administrator), the TOE overwrites the oldest audit record.

- Threshold of full audit trail: It can be set with an integer number between 90 and 99, in the unit of percent (%). The default value before setting is 90%.

SFR to be satisfied: FAU_STG.1, FAU_STG.3, FAU_STG.4

6.1.4. Audit data view and review

The audit data generated in the TOE are stored in SOHA Database which is an audit data storage, and the stored data are kept in a form of audit records in a table. The TOE provides the authorized administrator with the function to access the administrator interface on Petra Cipher Key Server via a web browser where the administrator uses functions to view and review the stored audit data on the administrator interface screen.

The TOE provides the GUI to view the audit data specified in the following [Table 6-2] Criteria for selection by audit data type, and does not provide the function to delete the audit data.

Audit Data Type	Audit Data Column Name	Criteria for Logical Relations
Administrator task history	Date and time of the task	Range search
	Category, Menu, Function, Content, ID, IP	AND search
Server history	Date and time of the task	Range search
	Content, Log	AND search
Encryption Key Request history	Date and time of the task	Range search
	Encrypt Key Id, Encrypt Key name, Algorithm, Encrypt mode, Request IP	AND search
Encryption task history	Date and time of the task	Range search
	DB, DB User, Schema, Table, Column, user IP, access program	AND search

[Table 6-2] Criteria for selection by audit data type

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

6.2. Cryptographic support

The TOE manages cryptographic keys for DB encryption and performs cryptographic operations. In addition, the TOE manages cryptographic keys and performs cryptographic operations for the encryption of the stored or transmitted TSF data

6.2.1. Cryptographic key generation and cryptographic operation

The TOE generates keys for the encryption of user data and keys for the encryption of TSF data by using the validated cryptographic module KLIB V3.0

Classification	Description
Cryptographic Module Name	KLIB V3.0
Developer	Korea University
Validation Date	August 6, 2021
Validation Level	General level: Security Level 1 Level by Item - Cryptographic module specification Level 1

	- Cryptographic module interface Level 1 - Roles, services and authentication Level 1 - Software/firmware Security Level 1 - Operational environment Level 1 - Physical security N/A - Non-invasive security N/A - Critical security parameter management Level 1 - Self-test Level 1 - Life cycle assurance Level 1 - Response to other attacks Level 1
Validation No.	CM-189-2026.8

[Table 6-3] Information on the validated cryptographic module used in the TOE

A random bit generator (HASH_DRBG SHA-256) provided by the validated cryptographic module KLIB V3.0 is used to generate an encryption key. [Table 6-4], [Table 6-5], and [Table 6-6] below explain cryptographic algorithms of the TOE, encryption key length and their uses.

List of Standards	Cryptographic Key Generation Algorithm	Cryptographic Key Length	Use
NIST SP 800-90	HASH_DRBG SHA-256	N/A	Cryptographic key generation

[Table 6-4] Cryptographic key generation algorithm and list of standards

List of Standards	Cryptographic Operation Algorithm	Cryptographic Key Length	Use
KS X 1213-1	ARIA (CBC, OFB, CFB)	128	User data encryption/decryption (symmetric key encryption)
	ARIA (CBC, OFB)	192	
	ARIA (CBC, OFB)	256	
TTAS.KO-12.0004/R1 ISO/IEC 18033-3	SEED (CBC, OFB, CFB)	128	User data encryption/decryption (symmetric key encryption)
ISO/IEC 10118-3 ISO/IEC 10118-3 Amd 1	SHA-256	N/A	User data encryption (one-way encryption)
	SHA-384	N/A	
	SHA-512	N/A	

[Table 6-5] User data cryptographic operation algorithm and list of standards

List of Standards	Cryptographic	Cryptographi	Use

	Operation Algorithm	c Key Length	
KS X 1213-1	ARIA (CBC)	256	Encryption of private key used for mutual authentication
			Encryption of private key used for encrypted communication
			Encryption of passwords of the authorized administrator
			Audit log encryption (Accessed administrator ID, accessed administrator IP, details)
			Encryption of TSF data cryptographic key
			Encryption of user data cryptographic key
ISO/IEC 14888-2	RSA-PSS	2048	Encryption of digital signature for mutual authentication
TTAS.KO-12.0004/R1 ISO/IEC 18033-3	SEED (CBC)	128	Encryption of the TOE internal communication data
ISO/IEC 10118-3 ISO/IEC 10118-3 Amd 1	SHA-256	N/A	Encryption of password of authorized administrator
			TSF data encryption key Integrity check value
			User data encryption key Integrity check value
ISO/IEC 9797-2	HMAC-SHA256	256	Generating an Encryption Key for the Master Key Using the Product Installation Password

[Table 6-6] TSF data cryptographic operation algorithm and list of standards

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

6.2.2. Cryptographic key distribution

The TOE generates a public/private key in advance to distribute an encryption key between TOE components, and exchanges a public key. It encrypts an encryption key to be distributed with the counterpart's public key that has been exchanged, and sends it. The counterpart that receives the key

decrypts it with its own private key to receive the encryption key.

An algorithm used in this process is RSAES 2048 bits provided by the validated cryptographic module KLIB V3.0 whose security and implementation conformance have been validated by Korea Cryptographic Module Validation Program (KCMVP).

List of Standards	Cryptographic Key Distribution Algorithm	Cryptographic Key Length	Use
ISO/IEC 18033-2	RSAES	2048	Cryptographic key distribution

[Table 6-7] Cryptographic key distribution algorithm and list of standards

SFR to be satisfied: FCS_CKM.2

6.2.3. Cryptographic key destruction

If an encryption key loaded on the memory expires when performing key generation, distribution and other tasks, all the bits used in the encryption key are overwritten with 0 to destroy the encryption key.

Key to be Destroyed	Type of Storage of Encryption Key	Destruction Method	Timing of Destruction
User data encryption key	Stored in DB	Overwrite with 0 three times	When the authorized administrator deletes an encryption key through the management function
Public key/private key	Memory	Overwrite with 0 three times	When a process using a public key/private key is terminated, or
Cryptographic key for encrypted communication	Memory	Overwrite with 0 three times	When the encrypted communication is terminated
TSF data encryption key	Memory	Overwrite with 0 three times	When encryption/decryption operation is terminated

[Table 6-8] List of cryptographic key destruction

SFR to be satisfied: FCS_CKM.4

6.2.4. Random bit generation

The TOE generates random bits needed to generate an encryption key by using the random bit generator (HASH_DRBG SHA-256) of the validated cryptographic module KLIB V3.0 whose security and implementation conformance has been validated by Korea Cryptographic Module Validation Program (KCMVP). Random bits are random values that are generated based on characters that combine the time information and program's internal address.

SFR to be satisfied: FCS_RBG.1(Extended)

6.3. User data protection

The TOE encrypts/decrypts the DB according to the policies established by the authorized administrator, and protects it from unauthorized disclosure by deleting the residual information after the encryption.

6.3.1. User data protection

The TOE protects the user data by providing the column-level encryption/decryption of the user data. In case of the plug-in type, Petra Cipher DB Agent performs the user data encryption/decryption at the column level. In case of the API type, Petra Cipher API Agent installed on the Application Server encrypts/decrypts user data. If the encryption/decryption of user data is completed, the original user data used are all deleted (by initializing the original user data to null). If a hash algorithm is used, only the encryption can be performed.

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

※ This is a requirement related to the function to encrypt/decrypt the user data. When the user data are encrypted, the same ciphertext is not generated for the same plaintext.

6.4. Identification and authentication

The TOE performs the identification and authentication to verify the identity of the authorized administrator, and provides the function to respond to a failed authentication. In addition, it performs the TOE internal mutual authentication.

6.4.1. Identification and authentication of the administrator

The TOE identifies and authenticates the administrator, and allows only one account for the authorized administrator role. The authorized administrator can change the ID and password provided by default after the initial login, and afterwards, can change the password only.

The TOE performs the identification and authentication based on the ID and password to verify the authorized administrator. Upon the initial login after the product is installed, the administrator shall change the ID and password. In this case, the password entered is output, being masked with "●" to protect the feedback. In case of failed authentication, the TOE does not provide feedback on the reason for failure, and outputs an error message that "Please check your ID or Password."

If identification and authentication attempts to authenticate the administrator fail consecutively (five times), the TOE locks the account. The identification and authentication are disabled for the locked account for five minutes, and the identification and authentication can be performed after five minutes based on the password entered.

Moreover, the TOE adds a unique value to a session ID generated when a web browser accesses the TOE, and maintains them. Then, if the session ID of the web browser is detected in another place, the TOE detects the reuse and blocks the session reusing the authentication data in order to prevent the reuse of the authentication data.

A password used in the TOE shall consist of the combination of numerical numbers between 9 and 13, English alphabets, and special characters. Characters that can be used for such combinations are shown in Defined Quality Metric in [Table 6-9] below.

Classification		Defined Quality Metric
Password combination rules	General combination rule	Passwords have a combination of three types of characters -alphabets, numbers and special characters. The length shall be at least 9 digits up to 13 digits.
	Number (10 numbers)	0-9
	Character (52 alphabets)	English upper case (26 alphabets) A-Z, English lower case (26 alphabets) a-z
	Special Characters (22 letters)	~ ! @ # $ % ^ & * _ + ` - { } : < > ? , . /

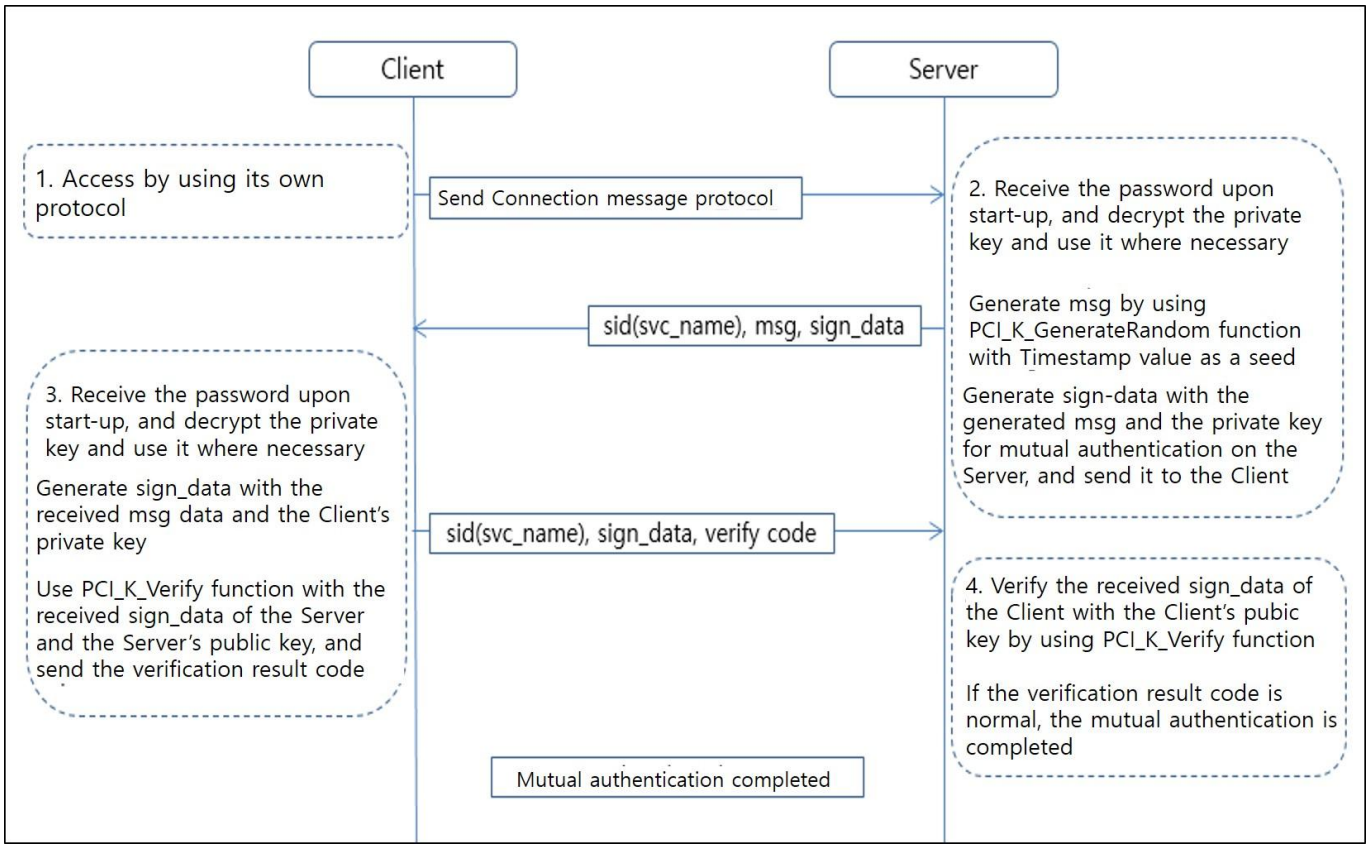
[Table 6-9] Password combination rules

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

6.4.2. TOE internal mutual authentication

The TOE provides the function of TOE internal mutual authentication through the validated cryptographic module, and uses the authentication protocol implemented by SINSIWAY Co., Ltd. The TOE generates a public key/private key by using RSA-PSS 2048bit (SHA-256) algorithm provided by the validated cryptographic module. A unique ID of a TOE component itself is used to generate a private key/public key, and the generated public key is exchanged between the TOE components, based on which the mutual authentication is performed.

The process of mutual authentication is described in the following diagram.



[Figure 6-1] Mutual authentication process

SFR to be satisfied: FIA_IMA.1

6.5. Security management

The TOE provides the security management function that enables the authorized administrator to configure and manage TOE security functions and TSF data. It allows only one account for the authorized administrator. The security management function provided is as follows:

Security Function Component	Management Function	Management Type
FAU_SAA.1	Modify information on the mail server that sends alarm mails, sender mails and receiver mails	Management of TSF data
FAU_SAR.1, FAU_SAR.3	View administrator task history, server history, encryption task history	Management of security function
FAU_STG.3	Modify the threshold value of the function to notify the threshold of the audit trail storage disk	Management of TSF data threshold value
FAU_STG.4	Modify the threshold value of the overwriting if the audit trail storage disk is full	
FAU_STG.3, FAU_STG.4	Modify the interval to check the capacity of the audit trail storage disk	Management of TSF data
FCS_CKM.1(1), FCS_CKM.4	Add/delete/modify an encryption key used for user data encryption/decryption	Management of security attributes
FMT_PWD.1	Change the password of the authorized administrator	Management of security function
FDP_UDE.1	Add/delete/modify information in the protected database when performing plug-in encryption	Management of security function
FDP_UDE.1	Collect table and column information in the protected database when performing plug-in encryption	Management of security function
FDP_UDE.1	Encrypt a column to be encrypted when performing plug-in encryption	Management of security function
FDP_UDE.1	Designate a column to be encrypted in the protected database when performing plug-in encryption	Management of security function

FDP_UDE.1	Register the library agent (Petra Cipher DB Agent, Petra Cipher API Agent) of the TOE and check the connection state when performing the encryption	Management of security function
FPT_TST.1	Verify the integrity	Management of security function
FTA_TSE.1	Add/modify allowed IP of the administrator PC which an authorized user can access	Management of TSF data

[Table 6-10] Security management function

The TOE enforces that the authorized administrator shall change the ID and password when he/she accesses the security management interface for the first time. The authorized administrator can change the administrator password through the security management interface. When the password of the authorized administrator is generated or changed, the TOE provides the following verification mechanism according to the password policy.

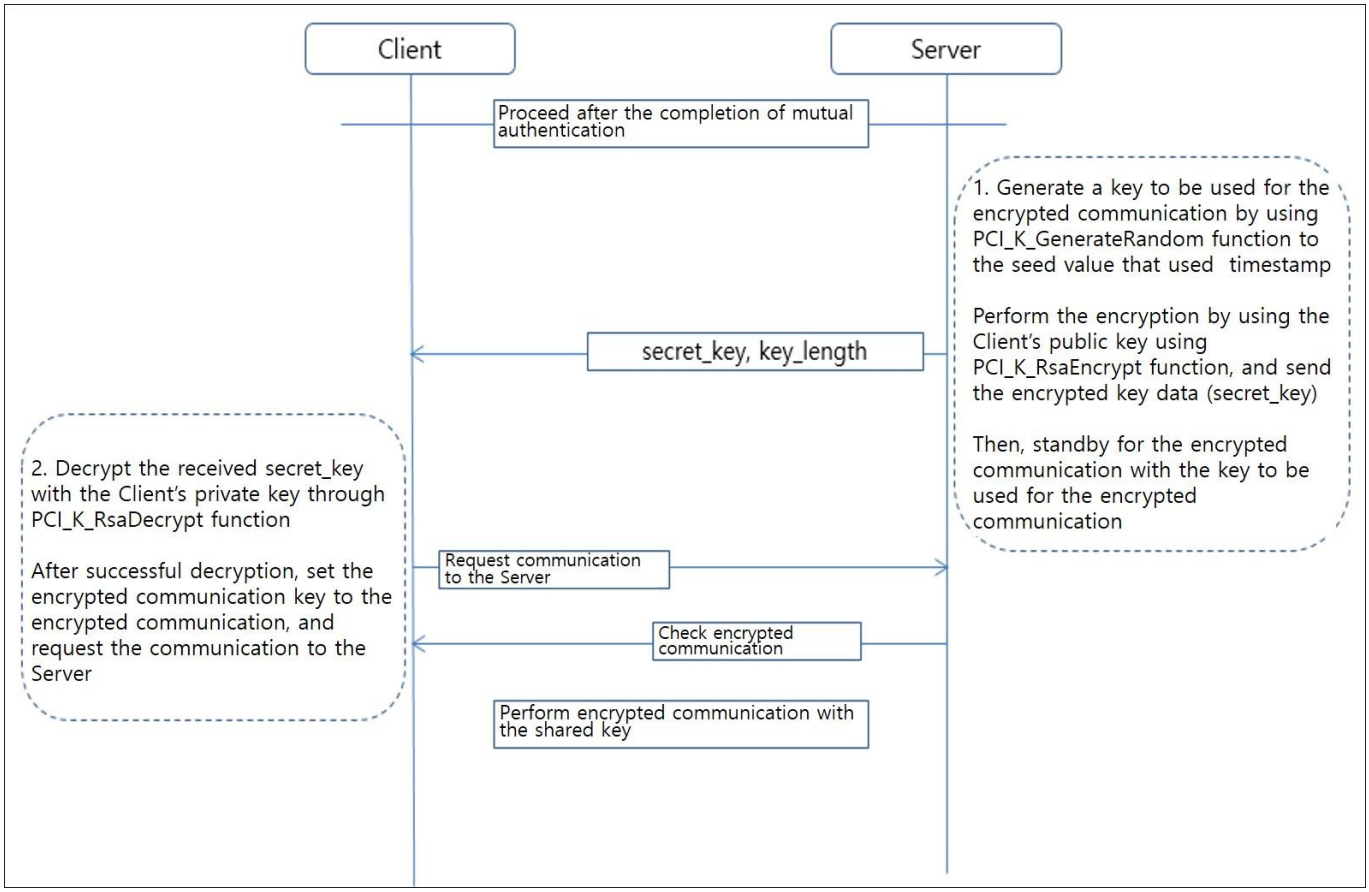
SFR to be satisfied: FMT_MOF.1, FMT_SMF.1, FMT_SMR.1, FMT_PWD.1

6.6. Protection of the TSF

6.6.1. Basic internal TSF data transfer protection

The TOE protects the internally transferred TSF data by using the validated protection function provided by the validated cryptographic module. It uses RSA-PSS SHA-256 algorithm to complete the mutual authentication based on the public key/private key (refer to 6.4.2 TOE internal mutual authentication).

Upon the completion of the mutual authentication, a random encryption communication key is generated, and distributed with RSAES SHA-256 algorithm, thereby providing the function of the encrypted communication between the TOE components. [Figure 6-2] below explains the encrypted communication between the TOE components.



[Figure 6-2] Encrypted communication between the TOE components

SFR to be satisfied: FPT_ITT.1

6.6.2. Basic protection of stored TSF data (Extended)

The encryption key, among the stored TSF data, is encrypted and protected with the master key in ARIA-256 CBC mode. The master key generates and uses a random value, based on characters that combine the time information and program's internal address with the password entered by the administrator.

In addition, the password entered by the administrator during the installation is used to encrypt and maintain the security policy and TOE setting values in ARIA-256 CBC mode. The administrator password is encrypted with SHA-256, and encrypted and stored in ARIA-256 CBC mode.

The list of the TSF data encrypted and managed is as follows:

Cryptographic Operation Algorithm	Cryptographic Key Length	Use	Use	Type of Storage

ARIA (CBC)	256	Encryption of private key used for mutual authentication	Encryption, Decryption	File
		Encryption of private key used for encrypted communication	Encryption, Decryption	File
		Encryption of passwords of the authorized administrator	Encryption, Decryption	Stored in DB
		Audit log encryption (Accessed administrator ID, accessed administrator IP, details)	Encryption, Decryption	Stored in DB
		Encryption of TSF data cryptographic key	Encryption, Decryption	Stored in DB, File
		Encryption of user data cryptographic key	Encryption, Decryption	Stored in DB
RSA-PSS	2048	Encryption of digital signature for mutual authentication	Encryption, Decryption	Memory
SEED (CBC)	128	Encryption of the TOE internal communication data	Encryption, Decryption	Memory
SHA-256	N/A	Encryption of passwords of the authorized administrator	One way encryption	Stored in DB
		TSF data encryption key Integrity check value	One way encryption(comparison of hash)	Stored in DB, File
		User data encryption key Integrity check value	One way encryption(comparison of hash)	Stored in DB
HMAC-SHA256	256	Generating an Encryption Key for the Master Key Using the Product Installation Password	Generating KEK	Memory

[Table 6-11] Encrypted TSF data and cryptographic operation algorithm

Type of Storage	Cryptographic Key Generation Algorithm	Cryptographic Key Length	Use
Memory	HASH_DRBG SHA-256	N/A	Cryptographic key generation

[Table 6-12] Encrypted TSF data and cryptographic key generation algorithm

Type of Storage	Cryptographic Key Distribution Algorithm	Cryptographic Key Length	Use
Memory	RSAES	2048	Cryptographic key distribution

[Table 6-13] Encrypted TSF data and cryptographic key distribution algorithm

SFR to be satisfied: FPT_PST.1(Extended)

6.6.3. TSF testing

The TOE performs self-tests (self-tests of the validated cryptographic module, integrity verification of the executable file and configuration of the TOE components, and process state check) upon the start-up of each component. Self-tests are performed on a periodic basis (60 seconds) after the start-up, and the self-test results are stored on Petra Cipher Key Server. If a self-test fails, the TOE component creates an audit log, and sends an alarm to the email set by the administrator. Furthermore, the authorized administrator can perform the integrity verification of executable files and configuration files of the TOE component by accessing Petra Cipher Key Server via a web browser.

TOE Classification	Self-test Item	Test Description
Petra Cipher Key Server	Validated cryptographic module	Self-test inside the validated cryptographic module
	Process	Check whether main processes necessary for the operation of Petra Cipher Key Server are in normal operation, and then send the test result to the audit log
Petra Cipher DB Agent	Validated cryptographic module	Self-test inside the validated cryptographic module
	Process	Check whether main processes necessary for the operation of Petra Cipher DB Agent are in normal operation, and then send the test result to the audit log
Petra Cipher API Agent	Validated cryptographic module	Self-test inside the validated cryptographic module
	Process	Check whether main processes necessary

		for the operation of Petra Cipher API Agent are in normal operation, and then send the test result to the audit log

[Table 6-14] TOE self-test items

TOE Classification	Integrity Test Item	Test Description
Petra Cipher Key Server	Executable file, configuration file	Check whether the executable file of Petra Cipher Key Server was corrupted by an unauthorized user, and the send the test result to the audit log
Petra Cipher DB Agent	Executable file, configuration file	Check whether the executable file of Petra Cipher DB Agent was corrupted by an unauthorized user, and the send the test result to the audit log
Petra Cipher API Agent	Executable file, configuration file	Check whether the executable file of Petra Cipher API Agent was corrupted by an unauthorized user, and the send the test result to the audit log

[Table 6-15] TOE integrity test items

SFR to be satisfied: FPT_TST.1

6.7. TOE access

The TOE provides only one administrator account that can access Petra Cipher Key Server. The account is available only if the ID and password are changed when the authorized administrator accesses for the first time.

The TOE provides the function that allows access only by a designated administrator PC with the allowed IP. Up to two IPs can be designated as accessible IP, which can be modified by the authorized administrator on Petra Cipher Key Server.

If an administrator session that is already accessed exists, the TOE blocks new access of an administrator session. The TOE provides the function to terminate a session if the authorized administrator remains inactive for a specified period of time (10 minutes).

SFR to be satisfied: FTA_MCS.2, FTA_SSL5(Extended), FTA_TSE.1