PrivacyDB V3.0
Security Target
V1.1

2025.03.17

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

< Revision History >

Ver	Date	Author	Revision
V1.0	Oct 30, 2024	Hyein Kim	Initial release
V1.1	Mar 17, 2025	Hyein Kim	Version update

< Table of Contents >

< List of Figures >

< Table of Contents >

1. ST introduction

1.1. ST Reference

Classification	Description
Title	PrivacyDB V3.0 Security Target
ST Version	V1.1
Author	JaeYeong Choi, Senior Researcher, Corporate Research Institute of OWL Systems Inc
Publication Date	March 17, 2025
Common Criteria	Common Criteria for Information Technology Security Evaluation
Common Criteria Version	CC:2022 Revision 1 - Part 1: Introduction and general model, CC:2022 R1 (CCMB-2022-11-001, 2022.11.) - Part 2: Security functional components, CC:2022 R1 (CCMB-2022-11-002, 2022..11) - Part 3: Security assurance components, CC:2022 R1 (CCMB-2022-11-003, 2022.11.) - Part 4: Framework for the specification of evaluation methods and activities, CC:2022 R1 (CCMB-2022-11-004, 2022.11.) - Part 5: Pre-defined packages of security requirements, CC:2022 R1 (CCMB-2022-11-005, 2022.11.) - Errata and Interpretation for CC:2022 (Release 1) and CEM:2022 (Release 1), Version 1.1, CCMB-2024-07-002, 2024. 7.
Evaluation Assurance Level	EAL 1+ (PP Compliance)
Keywords	DB encryption, Encryption

[Table 1] ST Reference

1.2. TOE Reference

The components of this TOE are divided into the following four S/W.

Classification		Contents	Type	Distribution type
TOE Identification		PrivacyDB V3.0	-	Distribute as a CD
TOE Version		V3.0.0.4	-	
TOE Component	PrivacyKMS	PrivacyKMS V3.0.0.1 - PrivacyKMS_linux_64bit_V3.0.0.1.tar	S/W	
	PrivacyConsole	PrivacyConsole V3.0.0.1	S/W	

		- PrivacyConsole_x64_V3.0.0.1.zip		
	PrivacyEOC_API	PrivacyEOC_API V3.0.0.1 - PrivacyEOC_API_linux_64bit_V3.0.0.1.tar	S/W	
	PrivacyEOC_Plug-in	PrivacyEOC_Plug-in V3.0.0.1 - PrivacyEOC_Plug-in_linux_64bit_V3.0.0.1.tar	S/W	
Guidance		PrivacyDB V3.0 Preparative Procedures V1.1 - PrivacyDB V3.0 Preparative Procedures V1.1.pdf PrivacyDB V3.0 User Operational Guidance V1.1 - PrivacyDB V3.0 User Operational Guidance V1.1.pdf	Electronic documents(PDF)	
Developers		JaeYeong Choi, Senior Researcher and four other developers in Corporate Research Institute of OWL Systems Inc		

[Table 2] TOE Reference

1.3. TOE Overview

PrivacyDB V3.0(hereinafter 'TOE') performs the function to prevent unauthorized exposure of the information with database(hereinafter 'DB') encryption. Cryptographic keys are used to encrypt user data that is managed by the key management server and stored in the DB.

TOE's encryption target is a DB that is managed by the database management system (hereinafter 'DBMS') in the operating environment and This security target defines all data as user data before and after encryption is stored in the DB. Depending on the security policy of the organization that operates the TOE, some or all of the user's data can be encrypted.

1.3.1. TOE Type and Scope

TOE is provided in the form of software and provides an encryption/decryption function for each column of user data. TOE is a DB encryption product that supports both methods which are divided into 'Plug-in method' and 'API method' according to the perform location of encryption and decryption of user data.

The components of TOE consist of the key management server PrivacyKMS (hereinafter 'KMS') for encryption and policy management, the API and plug-in modules (PrivacyEOC_API and PrivacyEOC_Plug-in, hereinafter collectively referred to as 'EOC') for performing encryption and decryption, and Management tools (hereinafter 'PrivacyConsole') for setting administrator policies.

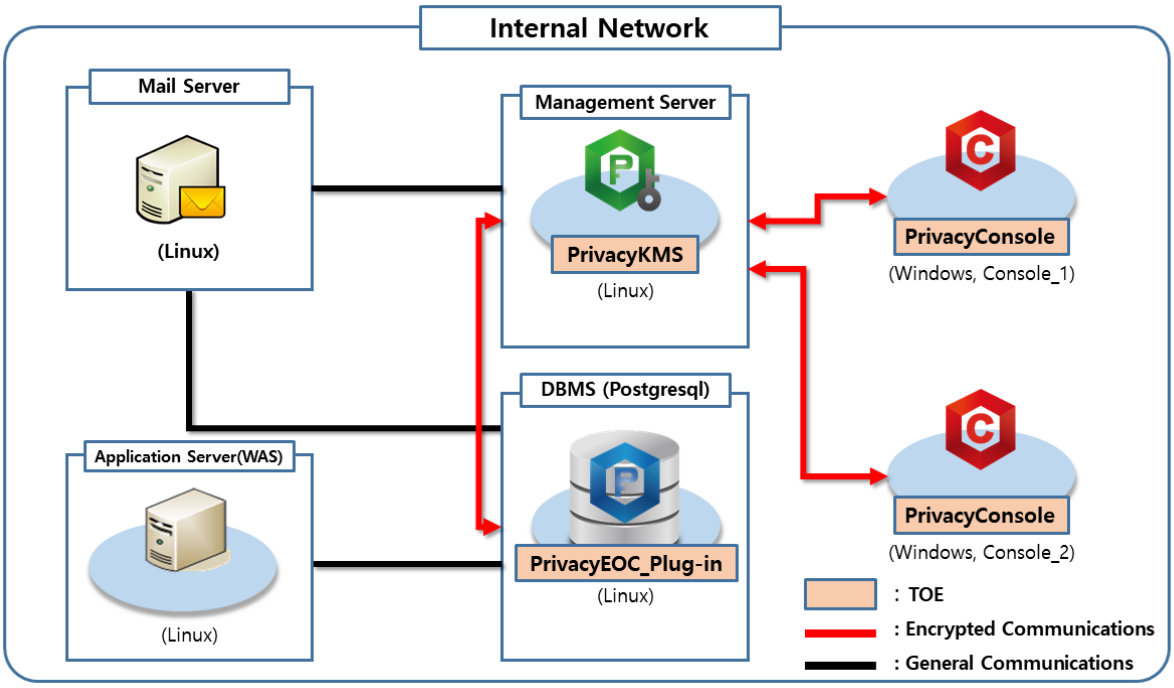
1.3.2. Usage and Major Security Features

The TOE is used to encrypt/decrypt the user data stored in the DB server according to the policy set by the authorized administrator to prevent the unauthorized disclosure of the confidential information.
TOE provides various security features such as the security audit function that records and manages major auditable events; cryptographic support function such as cryptographic key management to encrypt the user and the TSF data, and cryptographic operation; user data protection function that encrypts the user data and protects the residual information; identification and authentication function such as verifying the identity of the authorized administrator, authentication failure handling, and mutual authentication among the TOE components; security management function for security functions, role definition, and configuration; TSF protection functions including protecting the TSF data transmitted among the TOE components, protecting the TSF data stored in the storage that is controlled by the TSF, and TSF self-test; and TOE access function to manage the access session of the authorized administrator.

A Data Encryption Key (DEK) used to encrypt and decrypt user data is encrypted and protected with a Key Encryption Key (KEK). In addition, for communication between TOE components and protection of stored TSF data, the verification target encryption algorithm of the The validated cryptographic module(KCMVP) is used. Validated cryptographic modules used for user data encryption/decryption and storage data protection use OWLCrypto V1.0.

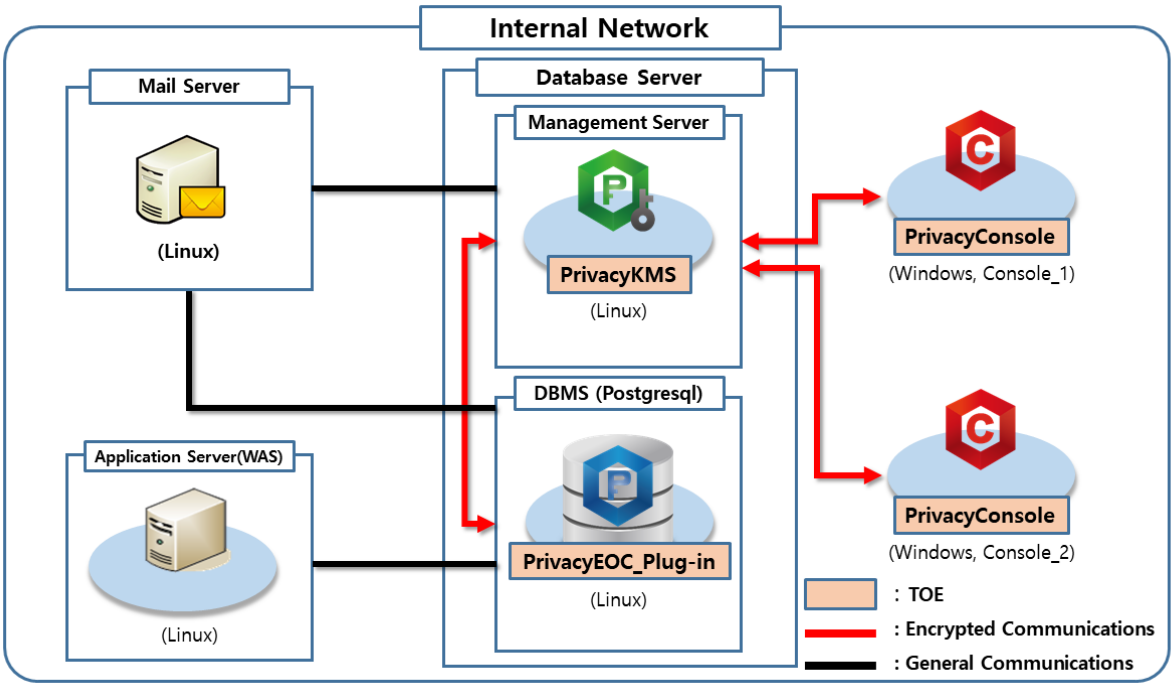
1.3.3. TOE Operational Environment

TOE's operating environment can be divided into a 'plug-in method' and an 'API method' as shown in the following figure, and includes a mail server for authorized administrator notifications when predicting potential security violations and audit data loss.
[Figure 1], [Figure 2] is a general plug-in operating environment. The EOC is installed within the Database Server where the protected DB resides and encrypts the user data received from the Application Server before storing it as DB in accordance with the security policy of an authorized administrator. EOC Performs the decryption of encrypted user data from the Database Server to the Application Server.



[Figure 1] Plug-in type operational environment (EOC, KMS separate type)

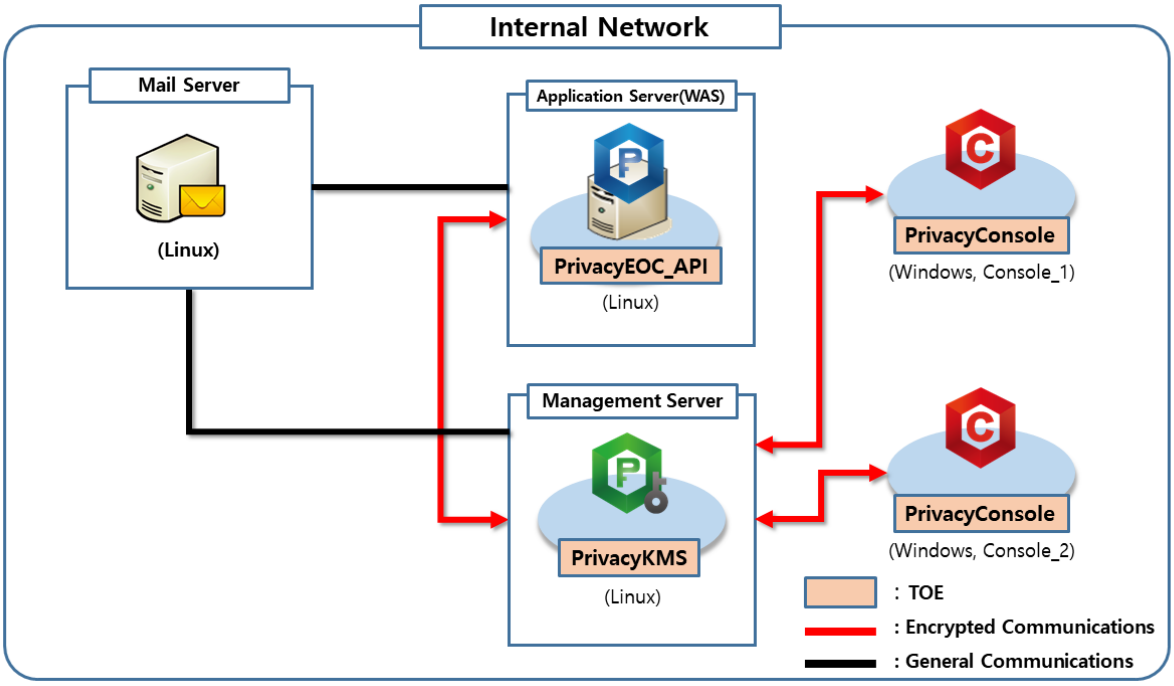
An authorized administrator accesses the KMS through the console to perform security management. An KMS can be installed with an EOC on a Database Server or physically separate from an EOC.



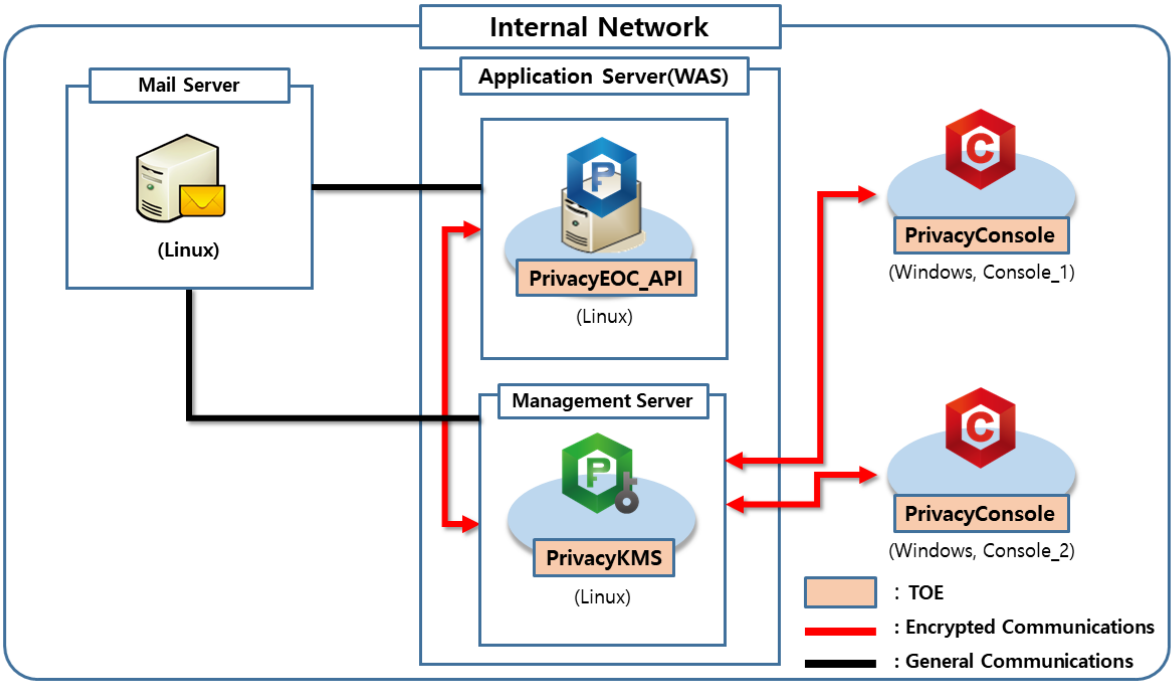
[Figure 2] Plug-in type operational environment (EOC, KMS intergrated type)

[Figure 3], [Figure 4] is a API type operating environment. The EOC is installed in Application Server and

perform encryption and decryption of user data in accordance with the security policies of an authorized administrator. The user data entered by the application user is encrypted by the EOC installed on the Application Server and sent to the Database Server. Encrypted user data from the Database Server is decrypted by the EOC installed on the Application Server and sent to the application user.
An authorized administrator accesses the KMS to perform security management. The KMS can be installed with the EOC in an Application Server or physically separate from the EOC



[Figure 3] API-type operational environment (EOC, KMS separate type)



[Figure 4] API-type operational environment (EOC, KMS integrated type)

1.3.4. Non-TOE environment required by TOE

In addition, the external IT entities required for TOE operations are:

- SMTP Server used to send alert mail to administrators

The hardware required for the TOE to be installed is as follows.

TOE Component	Minimum operation specification
PrivacyKMS	CPU: Intel Dual core 2.4 GHz or higher
PrivacyEOC_API	Memory: 8 GB Memory or higher
PrivacyEOC_Plug-in	HDD: Space required for TOE installation is 30 GB or higher
PrivacyConsole	NIC : 100/1000 Mbps * 1 EA or higher

[Table 3] Hardware minimum specifications

The 3rd Party software required for operation of the TOE is not included in the scope of the TOE, as follows:

TOE Component	Type	Contents	Notes
PrivacyKMS	OS	Ubuntu Pro 16.04.6 LTS 64 bit (GNU/Linux 4.4.0-142-generic x86_64)	
	DBMS	PostgreSQL 14.17	
PrivacyEOC_API	OS	Ubuntu Pro 16.04.6 LTS 64 bit (GNU/Linux 4.4.0-142-generic x86_64)	
PrivacyEOC_Plug-in	OS	Ubuntu Pro 16.04.6 LTS 64 bit (GNU/Linux 4.4.0-142-generic x86_64)	
	DBMS	PostgreSQL 14.17	DBMS to be protected
PrivacyConsole	OS	Windows 10 Pro 64 bit	
	JRE	Java JRE 8u421	

[Table 4] TOE Software requirements

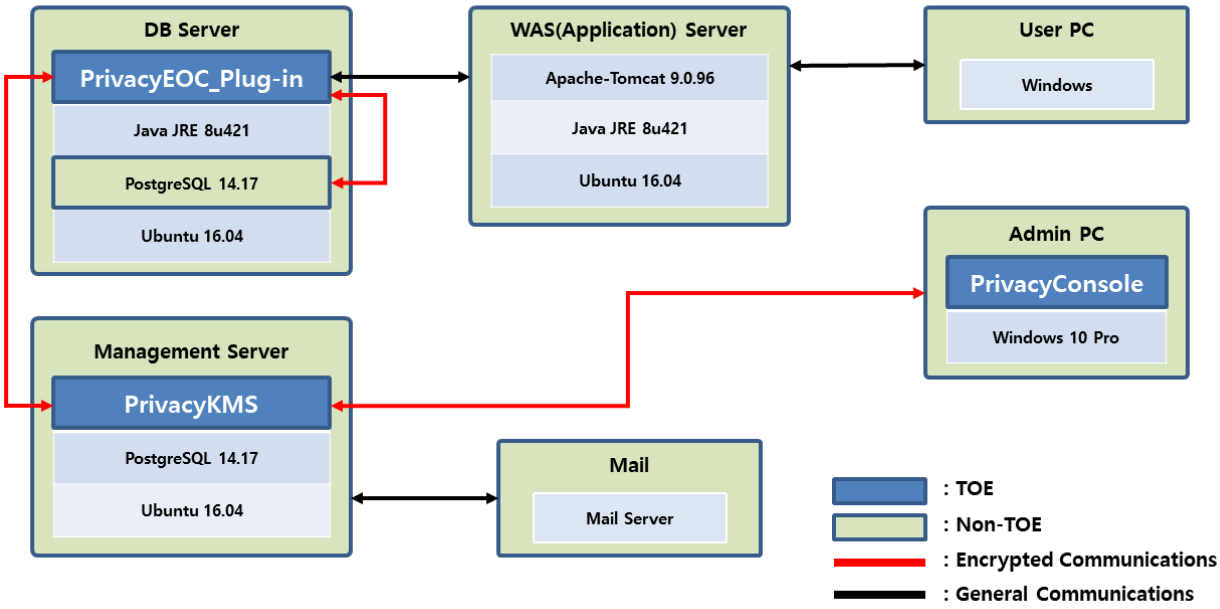
1.4. TOE Description

This section describes the physical and logical ranges of the TOE.

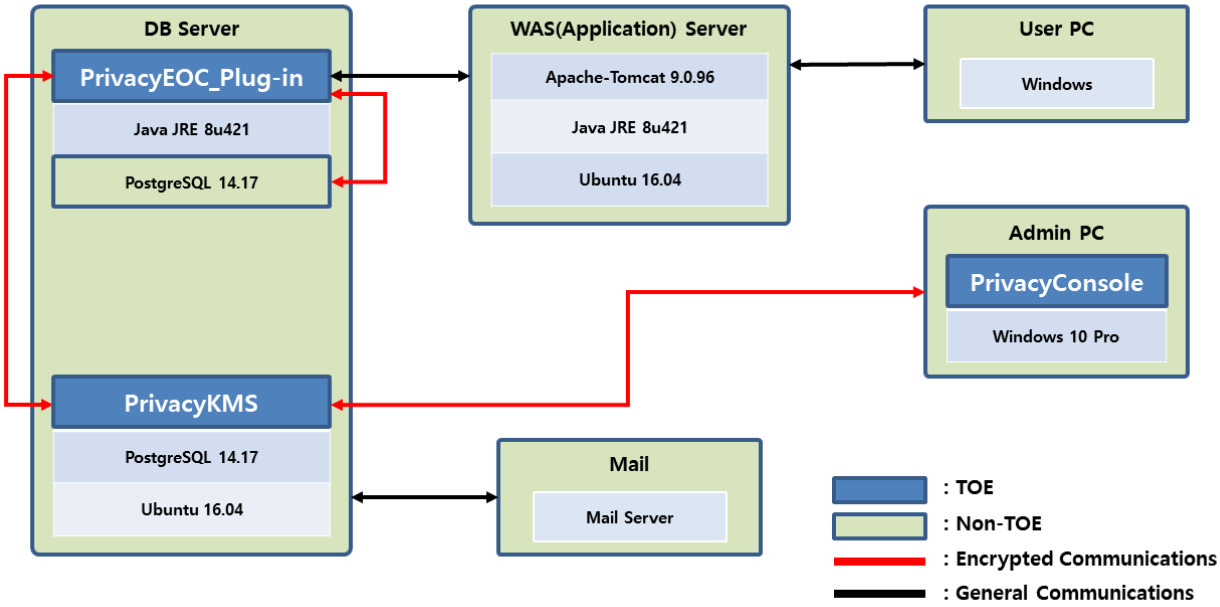
1.4.1. Physical Scope of the TOE

The TOE consists of KMS, which is security policy establishment and management server, Console, which

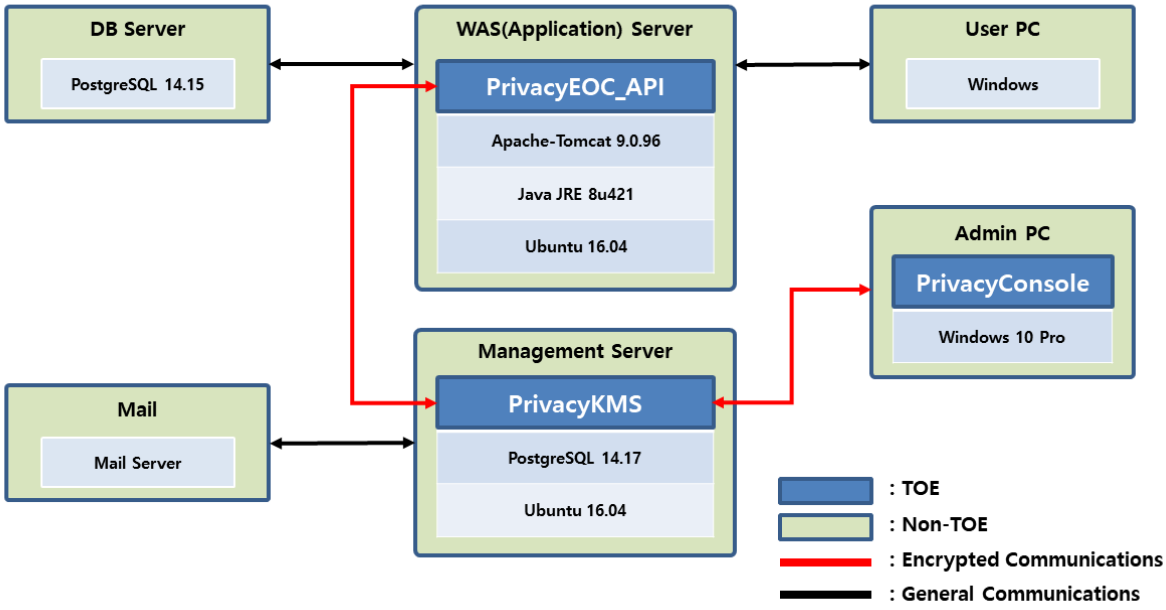
is an administrator tool, and EOC that encrypts and decrypts data in a DB or an application by receiving a DB cryptographic key and an encryption policy stored in KMS.
The EOC is installed on the WEB/WAS server for the API method and on the target DB server for the Plug-in method.



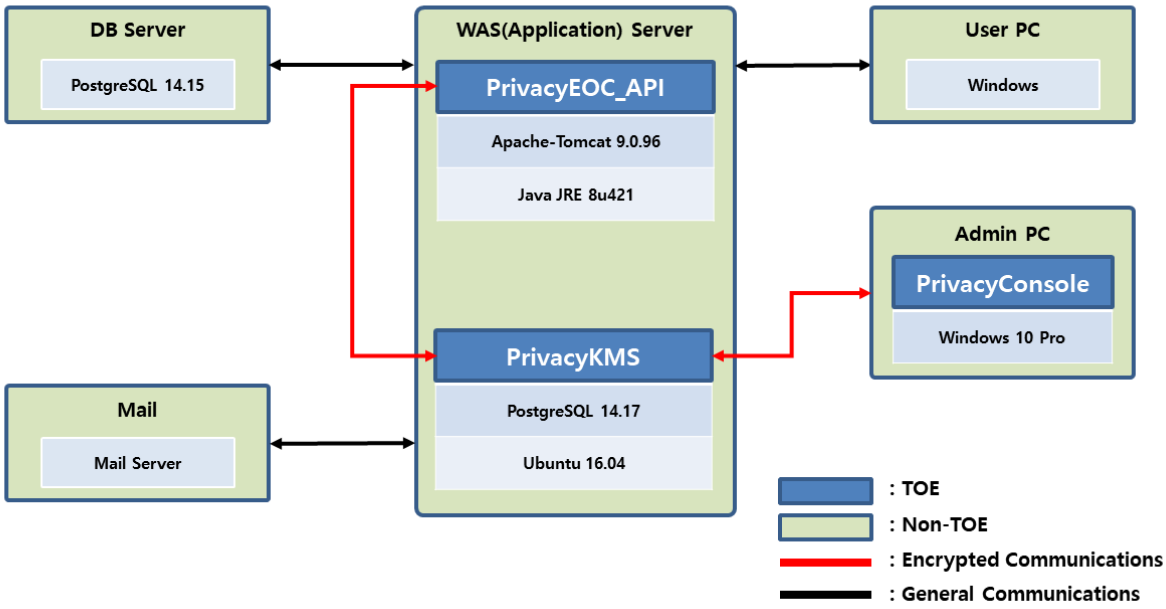
[Figure 5] TOE Physical Configuration - Plug-in (EOC, KMS separate type)



[Figure 6] TOE Physical Configuration - Plug-in(EOC, KMS intergrated type)



[Figure 7] TOE Physical Configuration - API(EOC, KMS separate type)



[Figure 8] TOE Physical Configuration - API(EOC, KMS integrated type)

The TOE consists of KMS, EOC, Console and User Operational Guidance and Preparation Procedure.

Classification	Contents	Type	Distribution type

TOE Identification		PrivacyDB V3.0	-	
TOE Version		PrivacyDB V3.0.0.4	-	
TOE Component	PrivacyKMS	PrivacyKMS V3.0.0.1 - PrivacyKMS_linux_64bit_V3.0.0.1.tar	S/W	Distributed as a CD
	PrivacyConsole	PrivacyConsole V3.0.0.1 - PrivacyConsole_x64_V3.0.0.1.zip	S/W	
	PrivacyEOC_API	PrivacyEOC_API V3.0.0.1 - PrivacyEOC_API_linux_64bit_ V3.0.0.1.tar	S/W	
	PrivacyEOC_Plug-in	PrivacyEOC_Plug-in V3.0.0.1 - PrivacyEOC_Plug-in_linux_64bit_ V3.0.0.1.tar	S/W	
Guidance		PrivacyDB V3.0 Preparative Procedures V1.1 - PrivacyDB V3.0 Preparative Procedures V1.1.pdf PrivacyDB V3.0 User Operational Guidance V1.1 - PrivacyDB V3.0 User Operational Guidance V1.1.pdf	Electronic documents (PDF)	

[Table 5] Physical scope of the TOE

The 3rd party software included in TOE is as follows.

TOE Component	Type	Purpose	Version
PrivacyKMS	openssl	TSF Data Transfer	V3.4.1
PrivacyEOC_API	openssl	TSF Data Transfer	V3.4.1
PrivacyEOC_Plug-in	openssl	TSF Data Transfer	V3.4.1
PrivacyConsole	openssl	TSF Data Transfer	V3.4.1
	zlib1	Certificate Generator Compression Library	V1.3.1

[Table 6] The 3rd party software included in TOE

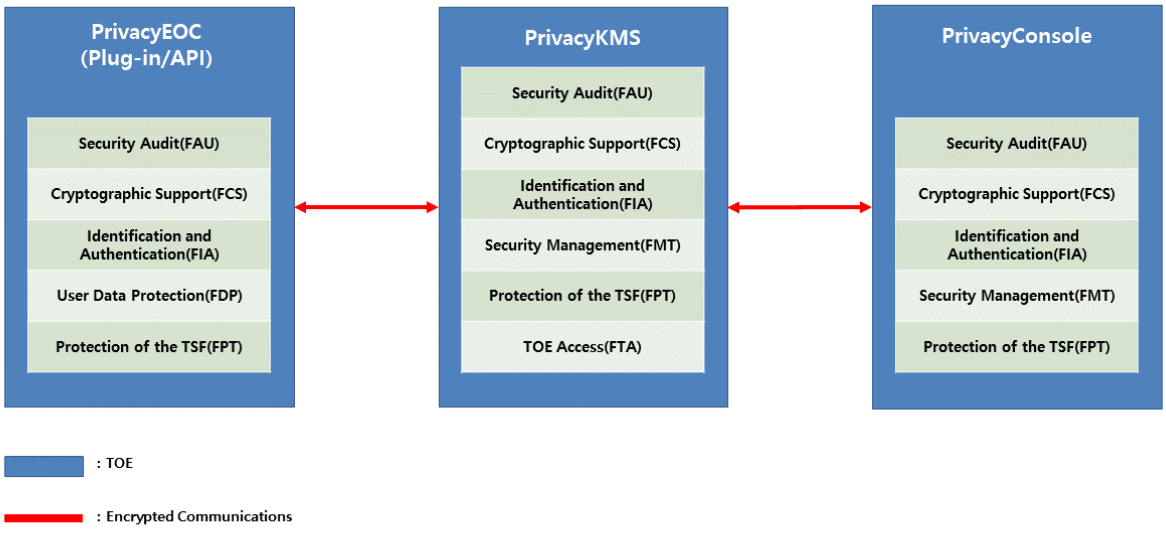
The KCMVP included in TOE is as follows.

Classification	Description
Cryptographic Module Name	OWLCrypto V1.0
Developed Company	OWL Systems Inc
Validation No	CM-241-2028.12
Module Type	S/W(Library)
Validation Date	Dec 22, 2023
Effective Expiration Date	Dec 22, 2028

[Table 7] KCMVP

1.4.2. Logical Scope of the TOE

The logical scope of the TOE consists of security audits, cryptographic support, user data protection, identification and authentication, security management, protection of the TSF, and TOE access, as shown in the [Figure 9] Logical scope of the TOE, as follows.



[Figure 9] Logical Scope of the TOE

1.4.2.1. PrivacyKMS

■ Security Audit

Each component (Console, EOC, KMS) generates audit data and transmits it to KMS, which stores the collected audit data in the DBMS. The audit data includes the date and time of the event, the type of the event, the identity of the subject, and the event result.

When generating audit data, KMS generate audit data for each event type (whether cipher text is included, encryption/decryption success log), and the generates an audit log for all audit data.

KMS sends an alert mail to the email address registered by the administrator when audit data is generated indicating potential security violations such as administrator continuous authentication failure, integrity violation, and KCMVP self-test failure.

The stored audit data can be inquired through the Console only by an authorized administrator. When inquiring audit data, Console can inquire audit data stored in DBMS through KMS after performing mutual authentication with KMS.

The KMS periodically monitors the audit data storage, and when the number of audit logs reaches 90% of the specified threshold, it generates an audit log indicating the threshold has been exceeded and sends

an alert email to authorized managers. When the audit data storage reaches 100% capacity, the KMS generates an audit log indicating storage saturation and sends an alert email to authorized managers. The KMS provides a function to overwrite the oldest audit data to ensure that the latest audit data is stored.

■ **Cryptographic Support**

TOE generates random numbers through a random number generator (HASH_DRBG (SHA256)) provided by the validated cryptographic module (KCMVP) and generates 256bit encryption keys for each component (Console, EOC, KMS). The generated encryption key is securely stored through the ARIA-256 block encryption algorithm.

KEK generation generates a 256 bit key through Password-Based Encryption Key Derivation (PBKDF) in accordance with the PKCS#5 standard. The following The validated cryptographic module(KCMVP) are used to generate keys.

Classification	Description
Cryptographic Module Name	OWLCrypto V1.0
Developed Company	OWL Systems Inc
Validation No	CM-241-2028.12
Module Type	S/W(Library)
Validation Date	Dec 22, 2023
Effective Expiration Date	Dec 22, 2028

[Table 8] KCMVP

Details of the The validated cryptographic module(KCMVP) that provides the random number generator used by KMS are as follows.

In the case of '/dev/urandom', the noise source output collection and configuration of the entropy source collects pseudo-random numbers provided by the operating system by opening the corresponding file and reading the data.

In the case of the 'time jitter' method, after intentionally increasing memory usage by using pseudo-random numbers, the difference between the timestamp counter values before and after memory usage is calculated. To perform the noise source health tests, Repetition Count Test
(RCT) and Adaptive Proportion Test (APT) are performed for all noise sources, respectively.

Since the finally collected seed has sufficient entropy, the conditioning process is omitted. The entropy source output configuration is collected in 48 bytes from each entropy source and is output in a binary data format.

From each entropy source data of 48 bytes that passed the noise source health test, only 32 bytes are obtained and connected to combine each entropy input. Simple connections are used as a combination method because each data provides sufficient entropy and independence is maintained regardless of the noise sources.

TOE distributes session keys for protecting transmission data among its components (Console, EOC, KMS). Transmission data protection is performed using AES_256_GCM_SHA384 with the distributed session key.

KMS induces a Key Encryption Key (KEK) from the password input according to the specified key derivation algorithm PBKDF conforming to TTAK.KO-12.0334-Part1/2 (2018) and the specified encryption key length of 256 bits.

After using the key encryption key (KEK), KEK derivation password, user data encryption key, TSF data encryption key, and certificate private key, KMS overwrites the encryption key memory area three times with "0" and releases the memory to safely destroy it.

When encrypting and decrypting user data stored in the DBMS that TOE wants to protect, the encryption operation is performed using ARIA-256 of the The validated cryptographic module(KCMVP). In addition, it provides user data encryption using a one-way encryption algorithm such as SHA-256, 512.

TSF data may also be encrypted using ARIA-256. Integrity shall be verified using RSA-PSS and SHA-256. During mutual authentication, certificate signatures shall be verified via RSA-PSS. Administrator passwords shall be hashed using SHA-256. A summary of cryptographic algorithms, key lengths, operational modes, and cryptographic operations is presented in the following table.

Classification	Standard	Algorithm	Key Length	Operation Mode	Operation List
User Data Encryption	KS X 1213-1	ARIA	256	CBC	DB Stored User Data En/Decryption
	ISO/IEC 10118-3	SHA-256, SHA-512	N/A	N/A	DB Stored User Data Encryption
TSF Data Encryption	KS X 1213-1	ARIA	256	CBC	Policy File, Transfer Data, DB CipherKey En/Decryption
	ISO/IEC 18033-2	RSAES	2048	N/A	Distribution CipherKey
	ISO/IEC 14888-2	RSA-PSS	2048	N/A	Mutual Authentication, Module and Configuration Integrity verification
	ISO/IEC 10118-3	SHA-256	N/A	N/A	Policy File Intergrity Verification,

					Administrator password encryption

[Table 9] TOE Encryption Operation

■ Identification and Authentication

The password required for manager authentication must be composed of a combination of at least 10 digits, 40 characters or less, numbers, uppercase letters (English), lowercase letters (English), and special characters according to the predefined combination rules to successfully authenticate. Setting a password that is identical to the user account (ID) is prohibited. The repeated use of the same character or number in succession is not allowed. Sequential input of characters or numbers based on keyboard layout is also prohibited. Reusing the most recently used password is not allowed either. In addition, in order to prevent the reuse of authentication data while the manager is authenticated, the sequence number is used to verify and prevent the reused data.

The TOE performs mutual authentication among its components KMS, EOC, and Console using a certificate-based proprietary authentication protocol.

After basic protection of internal TSF data transmission is established through TLS, EOC_API, EOC_Plug-in, and Console request the certificate from KMS, and KMS provides its certificate to each of these components.

Each component (EOC_API, EOC_Plug-in, and Console) verifies the received KMS certificate using a locally stored Root certificate to determine its trustworthiness.

If the KMS certificate is successfully validated, each component sends its own certificate to KMS.

KMS then verifies the received certificates from the components using the same Root certificate.

Once verification is successfully completed, mutual authentication is established.

■ Security Management

TOE succeeds in authentication based on ID and PW through Console, and only authorized managers who are logged in can perform security management functions.

When the administrator accesses for the first time, it is forced to change the password for the administrator, and the administrator can only perform security management through PrivacyConsole.

The security functions and management behavior capabilities that an authorized manager can manage are as follows.

Administrator Type	Management Type	Security Function	Management Behavior Capability			
			Determination for Behavior	Stop	Start	Change for Behavior
Authorized Administrator	Encryption Key	Creation an encryption/decryption	○	–	○	–

		key				
	Security Policy	Cipher Target Type	○	○	○	○
		Cipher Algorithm Type	○	○	○	○
		User Data Integrity Check Function	○	○	○	○
		Double Encryption	○	○	○	○
		Encryption Pattern	○	○	○	○
	Access Authority	User Access Authority	○	○	○	○
	Configuration	User Access Authority Allow / Reject Policy	○	○	–	○
		Administrator IP Setting	–	○	○	○
		Mail Server Setting	–	○	○	○
	Audit Log	Determining Audit Target(Ciphertext)	○	○	○	○
		Generating Success Log	○	○	○	○

[Table 10] The security functions and management behavior capabilities that an authorized manager can manage

Also, the types and management capabilities of TSF data managed by authorized administrators are as follows.

Administrator Type	Management Type	Data Type	Management Behavior Capability				
			Changing Defaults	Inquiry	Change	Creation	Deletion
Authorized Administrator	Key	User Data En/Decryption Key	–	○	–	○	○
		Master Key	–	–	○	○	–
	User	DB User	–	○	○	○	○
	Security Policy	User Data Cryptographic	○	○	○	○	○

		Policy					
	Access Authority	Access Time	○	○	–	○	○
		Access User	○	○	–	○	○
		Access IP	–	○	○	○	○
		Access Application	–	○	○	○	○
	Configuration	Admin IP	–	○	○	○	○
		Mail Server	–	○	○	○	○
	Audit Log	Admin Log	–	○	–	–	–
		Encryption Log	–	○	–	–	–
	Authentication Information	Password	–	–	○	○	–

[Table 11] The types and management capabilities of TSF data managed by authorized administrator

Guide at initial setting to set the password to a length of at least 10 digits with a combination of English/numerical/special characters, and force a reset to pop-up in case of violating this rule.

[Password Combination Rule]
- Digit : more than 10 ~ less then 40 character
- numbers, uppercase letters (English), lowercase letters (English), special character include at least on each
- Prohibit setting for User Account(ID) and the same Password
- Prohibit enter the same character or number repeatedly
- Prohibit sequential input of consecutive characters or numbers on the keyboard,
- Prohibit reuse previous password.
- Number(10) : 0~9,
- Uppercase Letters(26) : A~Z,
- Lowercase Letters(26) : a~z,
- Special Character(32) : `~!@#$%^&*()-_+=[]{}₩|;:'",.<>/?

■ Protection of the TSF

Cryptographic communication between TOE components(Console, EOC, KMS) uses the TLS 1.2 standard protocol to ensure safety, and the OpenSSL 3.4.1 library is applied to implementations. This process complies with the RFC 8446 standard.

AES_256_GCM_SHA384 256 bit key is used to ensure confidentiality and data integrity in cryptographic communication. The ECDHE algorithm is used in the key exchange process.
EOC_API, EOC-Plug-in, and Console exchange ephemeral keys with KMS using ECDHE (Elliptic Curve Diffie-Hellman Ephemeral), based on which a session key is derived.
During the data transmission phase, the generated session key is used together with the AES_256_GCM_SHA384 algorithm to perform data encryption/decryption and verify data integrity.

Among the stored TSF data, the DB encryption key is securely encrypted (ARIA-256) and stored by the master key protected by the KMS. The master key is securely stored by encrypting with ARIA-256 using an encryption key derived from the user password, and is used to encrypt and store policy files.
The administrator password is encrypted with SHA-256 and stored in the DB, and the encryption key is overwritten three times with '0' to destroy it safely after use, so it does not exist as a plaintext.
When storing TOE settings, encryption keys, and key security parameters, encrypt with TSF Data Encryption Key (ARIA-256)

Whenever KMS is driven, the password required for the key derivation algorithm PBKDF is input and the Master Key(DEK) is decrypted with the derived Key Encryption Key (KEK).

When KMS is driven, KMS performs its own self-test of the The validated cryptographic module(KCMVP) periodically (24 hours cycles) and performs its own test on the main KMS process. While the KMS is running, it periodically (24 hours cycles) performs its own self-test to see if the KMS process is running normally.
The KMS performs an integrity check function using the RSA-PSS algorithm on binary files such as executable files, configuration files, and library files at startup, periodically (every 24 hours), and upon administrator request.
If the noise source health tests fail during self-tests of the validated cryptographic module, it may be due to a temporary fault in the noise source. In such cases, a retry or reinitialization of the cryptographic module can be used to maintain the safe state of the random number generator.
If the TSF's own self-test fails, the authorized administrator will be notified by email.

■ TOE Access

TOE can only perform security management functions by an authorized administrator who identified and authenticated through the Console. The number of simultaneous sessions is limited to one authorized administrator based on the unique identification information and certificate of the administrator PC in which the Console is installed.
If an authorized administrator logged in through the Console has no input for 10 minutes, the session between the KMS and the Console is terminated and an administrator logout is performed automatically.

1.4.2.2. PrivacyEOC

■ **Security Audit**

EOC generates audit data and transmits it to KMS.

EOC sends an alert mail to the email address registered by the administrator when audit data is generated indicating potential security violations such as administrator continuous authentication failure, integrity violation, and KCMVP self-test failure.

■ **Cryptographic Support**

EOC generates a random number through a random number generator (HASH_DRBG (SHA256) provided by the (KCMVP) and generates encryption keys of 256 bits selectively according to the length of the encryption key selected by the administrator. The generated encryption key is securely stored through the ARIA-256 block encryption algorithm.

KEK generation generates a 256 bit key through Password-Based Encryption Key Derivation (PBKDF) in accordance with the PKCS#5 standard. The following The validated cryptographic module(KCMVP) are used to generate keys.

Classification	Description
Cryptographic Module Name	OWLCrypto V1.0
Developed Company	OWL Systems Inc
Validation No	CM-241-2028.12
Module Type	S/W(Library)
Validation Date	Dec 22, 2023
Effective Expiration Date	Dec 22, 2028

[Table 12] KCMVP

Details of the The validated cryptographic module(KCMVP) that provides the random number generator used by TOE are as follows.

In the case of '/dev/urandom', the noise source output collection and configuration of the entropy source collects pseudo-random numbers provided by the operating system by opening the corresponding file and reading the data.

In the case of the 'time jitter' method, after intentionally increasing memory usage by using pseudo-random numbers, the difference between the timestamp counter values before and after memory usage is calculated. To perform the noise source health tests, Repetition Count Test
(RCT) and Adaptive Proportion Test (APT) are performed for all noise sources, respectively.

Since the finally collected seed has sufficient entropy, the conditioning process is omitted. The entropy source output configuration is collected in 48 bytes from each entropy source and is output in a binary

data format.

From each entropy source data of 48 bytes that passed the noise source health test, only 32 bytes are obtained and connected to combine each entropy input. Simple connections are used as a combination method because each data provides sufficient entropy and independence is maintained regardless of the noise sources.

EOC distributes session keys for protecting transmission data among TOE components(Console, EOC, KMS). Transmission data protection is performed through AES_256_GCM_SHA384 using the distributed session key.

EOC induces a Key Encryption Key (KEK) from the password input according to the specified key derivation algorithm PBKDF conforming to TTAK.KO-12.0334-Part1/2 (2018) and the specified encryption key length of 256 bits.

After using the key encryption key (KEK), KEK derivation password, user data encryption key, TSF data encryption key, and certificate private key, EOC overwrites the encryption key memory area three times with "0" and releases the memory to safely destroy it.

When encrypting and decrypting user data stored in the DBMS that TOE wants to protect, the encryption operation is performed using ARIA-256 of the The validated cryptographic module(KCMVP). In addition, it provides user data encryption using a one-way encryption algorithm such as SHA-256, 512.

When encrypting TSF transmission data, encryption/decryption is performed using the ARIA-256 algorithm of the The validated cryptographic module(KCMVP), and when encrypting TSF storage data, it can be encrypted with ARIA-256.

TSF data may also be encrypted using ARIA-256. Integrity shall be verified using RSA-PSS and SHA-256. During mutual authentication, certificate signatures shall be verified via RSA-PSS. A summary of cryptographic algorithms, key lengths, operational modes, and cryptographic operations is presented in the following table.

Classification	Standard	Algorithm	Key Length	Operation Mode	Operation List
User Data Encryption	KS X 1213-1	ARIA	256	CBC	DB Stored User Data En/Decryption
	ISO/IEC 10118-3	SHA-256, SHA-512	N/A	N/A	DB Stored User Data Encryption

TSF Data Encryption	KS X 1213-1	ARIA	256	CBC	Policy File, Transfer Data, DB CipherKey En/Decryption
	ISO/IEC 18033-2	RSAES	2048	N/A	Distribution CipherKey
	ISO/IEC 14888-2	RSA-PSS	2048	N/A	Mutual Authentication, Module and Configuration Integrity verification
	ISO/IEC 10118-3	SHA-256	N/A	N/A	Policy File Intergrity Verification, Administrator password encryption

[Table 13] TOE Encryption Operation

■ Identification and Authentication

Each component (EOC_API, EOC_Plug-in, and Console) verifies the received KMS certificate using a locally stored Root certificate to determine its trustworthiness.
If the KMS certificate is successfully validated, each component sends its own certificate to KMS.
KMS then verifies the received certificates from the components using the same Root certificate.
Once verification is successfully completed, mutual authentication is established.

■ User data protection

It provides a column-specific encryption/decryption method for user data stored in the DBMS that TOE wants to protect, and performs encryption/decryption in the Web Application Server or DBMS according to the API and Plug-In method.

After the TOE encrypts and decrypts the user data, the remaining information is overwritten with "0" three times on the memory to protect the remaining information on the original data, and the memory is released and completely destroyed.

■ Protection of the TSF

Cryptographic communication between TOE components(Console, EOC, KMS) uses the TLS 1.2 standard protocol to ensure safety, and the OpenSSL 3.4.1 library is applied to implementations. This process complies with the RFC 8446 standard.

AES_256_GCM_SHA384 256 bit key is used to ensure confidentiality and data integrity in cryptographic communication. The ECDHE algorithm is used in the key exchange process.
EOC_API, EOC-Plug-in, and Console exchange ephemeral keys with KMS using ECDHE (Elliptic Curve Diffie-

Hellman Ephemeral), based on which a session key is derived.
During the data transmission phase, the generated session key is used together with the AES_256_GCM_SHA384 algorithm to perform data encryption/decryption and verify data integrity.

Among the stored TSF data, the DB encryption key is securely encrypted (ARIA-256) and stored by the master key protected by the EOC. The master key is securely stored by encrypting with ARIA-256 using an encryption key derived from the user password, and is used to encrypt and store policy files.
When storing EOC settings, encryption keys, and key security parameters, encrypt with TSF Data Encryption Key (ARIA-256)
Whenever EOC is driven, the password required for the key derivation algorithm PBKDF is input and the Master Key(DEK) is decrypted with the derived Key Encryption Key (KEK).

When EOC is driven, EOC performs its own self-test of the The validated cryptographic module(KCMVP) periodically (24 hours cycles) and performs its own test on the main EOC process. While the EOC is running, it periodically (24 hours cycles) performs its own self-test to see if the EOC process is running normally.
The EOC performs an integrity check function using the RSA-PSS algorithm on binary files such as executable files, configuration files, and library files at startup, periodically (every 24 hours), and upon administrator request.
If the noise source health tests fail during self-tests of the validated cryptographic module, it may be due to a temporary fault in the noise source. In such cases, a retry or reinitialization of the cryptographic module can be used to maintain the safe state of the random number generator.
If the TSF's own self-test fails, the authorized administrator will be notified by email.

1.4.2.3. PrivacyConsole

■ **Security Audit**

Console generates audit data and transmits it to KMS.
Console sends an alert mail to the email address registered by the administrator when audit data is generated indicating potential security violations such as administrator continuous authentication failure, integrity violation, and KCMVP self-test failure.
The stored audit data can be inquired through the Console only by an authorized administrator. When inquiring audit data, Console can inquire audit data stored in DBMS through KMS after performing mutual authentication with KMS.
The audit data can be checked for detailed information in the log details through the Console's security audit interface, and the authorized manager can selectively inquire the accumulated audit data for each audit data type.

■ **Cryptographic Support**

Console distributes session keys for protecting transmission data among TOE components(Console, EOC, KMS). Transmission data protection is performed through AES_256_GCM_SHA384 using the distributed

session key.
TOE induces a Key Encryption Key (KEK) from the password input according to the specified key derivation algorithm PBKDF conforming to TTAK.KO-12.0334-Part1/2 (2018) and the specified encryption key length of 256 bits.

After using the KEK derivation password and certificate private key, TOE overwrites the encryption key memory area three times with "0" and releases the memory to safely destroy it.

When encrypting TSF transmission data, encryption/decryption is performed using the ARIA-256 algorithm of the The validated cryptographic module(KCMVP), and when encrypting TSF storage data, it can be encrypted with ARIA-256.

Integrity shall be verified using RSA-PSS and SHA-256. During mutual authentication, certificate signatures shall be verified via RSA-PSS. A summary of cryptographic algorithms, key lengths, operational modes, and cryptographic operations is presented in the following table.

Classification	Standard	Algorithm	Key Length	Operation Mode	Operation List
User Data Encryption	KS X 1213-1	ARIA	256	CBC	DB Stored User Data En/Decryption
	ISO/IEC 10118-3	SHA-256, SHA-512	N/A	N/A	DB Stored User Data Encryption
TSF Data Encryption	KS X 1213-1	ARIA	256	CBC	Policy File, Transfer Data, DB CipherKey En/Decryption
	ISO/IEC 18033-2	RSAES	2048	N/A	Distribution CipherKey
	ISO/IEC 14888-2	RSA-PSS	2048	N/A	Mutual Authentication, Module and Configuration Integrity verification
	ISO/IEC 10118-3	SHA-256	N/A	N/A	Policy File Intergrity Verification, Administrator password encryption

[Table 14] TOE Encryption Operation

■ Identification and Authentication

TOE provides identification and authentication functions based on the manager's ID and PW through the

Console, and additionally provides authentication functions through registered certificates. Only certificate generation can be performed before manager identification and authentication. If successive authentication failures (five consecutive times) occur by the administrator while performing authentication, the TOE restricts the login function of the ID for 5 minutes so that the authentication function is no longer possible. Console blocks information exposed on the screen by masking ('*') secret information such as passwords input during administrator authentication, and does not provide an accurate reason for authentication failure so that the password cannot be inferred through pop-up messages that occur when authentication fails.

The password required for manager authentication must be composed of a combination of at least 10 digits, 40 characters or less, numbers, uppercase letters (English), lowercase letters (English), and special characters according to the predefined combination rules to successfully authenticate. Setting a password that is identical to the user account (ID) is prohibited. The repeated use of the same character or number in succession is not allowed. Sequential input of characters or numbers based on keyboard layout is also prohibited. Reusing the most recently used password is not allowed either. In addition, in order to prevent the reuse of authentication data while the manager is authenticated, the sequence number is used to verify and prevent the reused data.

The TOE performs mutual authentication among its components KMS, EOC, and Console using a certificate-based proprietary authentication protocol.

After basic protection of internal TSF data transmission is established through TLS, EOC_API, EOC_Plug-in, and Console request the certificate from KMS, and KMS provides its certificate to each of these components.

Each component (EOC_API, EOC_Plug-in, and Console) verifies the received KMS certificate using a locally stored Root certificate to determine its trustworthiness.

If the KMS certificate is successfully validated, each component sends its own certificate to KMS.

KMS then verifies the received certificates from the components using the same Root certificate.

Once verification is successfully completed, mutual authentication is established.

■ **Security Management**

Console succeeds in authentication based on ID and PW through Console, and only authorized managers who are logged in can perform security management functions. The TOE provides security function management, TSF data management, and ID and password management functions, which are performed by the authorized administrator.

When the administrator accesses for the first time, it is forced to change the password for the administrator, and the administrator can only perform security management through Privacy Console.

The security functions and management behavior capabilities that an authorized manager can manage are as follows.

Administrator	Management	Security Function	Management Behavior Capability

Type	Type		Determination for Behavior	Stop	Start	Change for Behavior
Authorized Administrator	Encryption Key	Creation an encryption/decryption key	○	–	○	–
	Security Policy	Cipher Target Type	○	○	○	○
		Cipher Algorithm Type	○	○	○	○
		User Data Integrity Check Function	○	○	○	○
		Double Encryption	○	○	○	○
		Encryption Pattern	○	○	○	○
	Access Authority	User Access Authority	○	○	○	○
	Configuration	User Access Authority Allow / Reject Policy	○	○	–	○
		Administrator IP Setting	–	○	○	○
		Mail Server Setting	–	○	○	○
	Audit Log	Determining Audit Target(Ciphertext)	○	○	○	○
		Generating Success Log	○	○	○	○

[Table 15] The security functions and management behavior capabilities that an authorized manager can manage

Also, the types and management capabilities of TSF data managed by authorized administrators are as follows.

Administrator Type	Management Type	Data Type	Management Behavior Capability				
			Changing Defaults	Inquiry	Change	Creation	Deletion
Authorized Administrator	Key	User Data En/Decryption Key	–	○	–	○	○

		Master Key	–	–	○	○	–
	User	DB User	–	○	○	○	○
	Security Policy	User Data Cryptographic Policy	○	○	○	○	○
	Access Authority	Access Time	○	○	–	○	○
		Access User	○	○	–	○	○
		Access IP	–	○	○	○	○
		Access Application	–	○	○	○	○
	Configuration	Admin IP	–	○	○	○	○
		Mail Server	–	○	○	○	○
	Audit Log	Admin Log	–	○	–	–	–
		Encryption Log	–	○	–	–	–
	Authentication Information	Password	–	–	○	○	–

[Table 16] The types and management capabilities of TSF data managed by authorized administrator

Guide at initial setting to set the password to a length of at least 10 digits with a combination of English/numerical/special characters, and force a reset to pop-up in case of violating this rule.

[Password Combination Rule]
- Digit : more than 10 ~ less then 40 character
- numbers, uppercase letters (English), lowercase letters (English), special character include at least on each
- Prohibit setting for User Account(ID) and the same Password
- Prohibit enter the same character or number repeatedly
- Prohibit sequential input of consecutive characters or numbers on the keyboard,
- Prohibit reuse previous password.
- Number(10) : 0~9,
- Uppercase Letters(26) : A~Z,
- Lowercase Letters(26) : a~z,
- Special Character(32) : `~!@#$%^&*()-_+=[]{}₩|;:'",.<>/?

■ **Protection of the TSF**

Cryptographic communication between TOE components(Console, EOC, KMS) uses the TLS 1.2 standard protocol to ensure safety, and the OpenSSL 3.4.1 library is applied to implementations. This process complies with the RFC 8446 standard.

AES_256_GCM_SHA384 256 bit key is used to ensure confidentiality and data integrity in cryptographic communication. The ECDHE algorithm is used in the key exchange process.
EOC_API, EOC-Plug-in, and Console exchange ephemeral keys with KMS using ECDHE (Elliptic Curve Diffie-Hellman Ephemeral), based on which a session key is derived.
During the data transmission phase, the generated session key is used together with the AES_256_GCM_SHA384 algorithm to perform data encryption/decryption and verify data integrity.

Whenever KMS is driven, the password required for the key derivation algorithm PBKDF is input and the Master Key(DEK) is decrypted with the derived Key Encryption Key (KEK).
When Console is driven, Console performs its own self-test of the The validated cryptographic module(KCMVP) periodically (24 hours cycles) and performs its own test on the main Console process. While the Console is running, it periodically (24 hours cycles) performs its own self-test to see if the Console process is running normally.
The Console performs an integrity check function using the RSA-PSS algorithm on binary files such as executable files, configuration files, and library files at startup, periodically (every 24 hours), and upon administrator request.
If the noise source health tests fail during self-tests of the validated cryptographic module, it may be due to a temporary fault in the noise source. In such cases, a retry or reinitialization of the cryptographic module can be used to maintain the safe state of the random number generator.
If the TSF's own self-test fails, the authorized administrator will be notified by email.

1.5. Conventions

The notation, form, and writing rules used follow common evaluation criteria.

Common evaluation criteria allow iterative, assignment, selection, and refinement operations that can be performed in security function requirements. Each operation is used in this protection profile.

This security target specification uses the same rules for creating common evaluation criteria for selection, assignment, refinement, and repetition operations.

Iteration

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

Assignment

This is used to assign specific values to unspecified parameters (e.g., password length).

The result of assignment is indicated in square brackets like [assignment value].

Selection

This is used to select one or more options provided by the CC in stating a requirement.

The result of selection is shown as *underlined and italicized*.

Refinement

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

1.6. Terms and Definitions

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

Approved cryptographic algorithm

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

Application Server

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

Approved mode of operation

The mode of cryptographic module using approved cryptographic algorithm.

Self test

Pre-operational and conditional testing performed by the cryptographic module.

Assets

Entities that the owner of the TOE presumably places value upon

Assignment

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

Attack potential

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

Authorized Administrator

Authorized user to securely operates and manages the TOE

Authentication Data

Information used to verify the claimed identity of a user

Authorized User

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

Can/could

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

column

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

Component

Smallest selectable set of elements on which requirements may be based

Critical Security Parameters(CSP)

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

Class

Set of CC families that share a common focus

Database(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.

Database Server

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

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 ST, refers to the database management system based on the relational database model.

Data Encryption Key(DEK)

Key that encrypts and decrypts data.

DB encryption key

Key to encrypt and decrypt real table columns

Decryption

The act that restoring the cipher text into the plaintext using the decryption key

Dependency

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

Encryption

The act that converts the plaintext into the cipher text using the encryption key

Element

Indivisible statement of a security need

External Entity

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

Evaluation Assurance Level(EAL)

Set of assurance requirements drawn from CC Part 3, representing a point on the CC predefined assurance scale, that form an assurance package

Family

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

Identity

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

Iteration

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

Key Encryption Key (KEK)

Key that encrypts and decrypts another cryptographic key

Management access

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

Master encryption key

This is the key to encrypt master key

Master key

Key to encrypt the Encryption key when saving it to a file

Object

Passive entity in the TOE containing or receiving information and on which subjects perform operations

Operation(on a component of the CC)

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

Operation(on a subject)

Specific type of action performed by a subject on an object

Private Key

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

Protection Profile(PP)

Implementation-independent statement of security needs for a TOE type

Public Key

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

Public Key (Asymmetric) Cryptographic Algorithm

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

Random Bit Generator(RBG)

A device or algorithm that outputs a binary 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.

Recommend/be recommended

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

Refinement
Addition of details to a component

Role
Predefined set of rules establishing the allowed interactions between a user and the TOE

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

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

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

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

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

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

Session Key
Encryption key to encrypt and decrypt data in the communications section

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

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

Subject

Active entity in the TOE that performs operations on objects

Target of Evaluation(TOE)

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

TLS(Transport Layer Security)

An SSL-based cryptographic authentication communication protocol between servers and clients, described in RFC 2246.

TOE Security Functionality(TSF)

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

TSF Data

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

User

Refer to "External entity"

User Data

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

2. Declaration of compliance

2.1. Declaration of compliance of CC, PP, Package

CC, PP and Package that are compliant with ST and TOE are as follows.

.

Classification	Compliance
Declaration of compliance with the Common Criteria	Common Criteria for Information Technology Security Evaluation CC:2022 Revision 1 • Common Criteria for Information Technology Security Evaluation CC:2022 R1 Errata and Interpretation for CC:2022 (Release 1) and CEM:2022 (Release 1), Version 1.1, CCMB-2024-07-002, 2024. 7. • - Common Criteria for Information Technology Security Evaluation. Part 1: Introduction and General Model, Version CC:2022 R1 (CCMB-2022-11-001, 2022.11) • Common Criteria for Information Technology Security Evaluation. Part 2: Security Functional Components, Version CC:2022 R1 (CCMB-2022-11-002, 2022.11) • Common Criteria for Information Technology Security Evaluation. Part 3: Security Assurance Components, Version CC:2022 R1 (CCMB-2022-11-003, 2022.11) • Common Criteria for Information Technology Security Evaluation. Part 4: Framework for Evaluation Method and Activity Specification, Version CC:2022 R1 (CCMB-2022-11-004, 2022.11) • Common Criteria for Information Technology Security Evaluation. Part 5: Predefined Security Requirements Package, Version CC:2022 R1 (CCMB-2022-11-005, 2022.11.)
Declaration of compliance with the Common Criteria Part 2	Part2 Extended: FIA_IMA.1, FDP_UDE.1, FMT_PWD.1, FPT_PST.1
Declaration of compliance with the Common Criteria Part 3	*Conformant*
Protection Profile	Korean National PP for Database Encryption V3.0
Declaration of Package	Augmented: EAL1+ *augmented* (ATE_FUN.1)

[Table 17] CC Declaration of compliance

2.2. Compliance Method

2.2.1 Reference to evaluation methods/activities

This security target specification uses the evaluation method/evaluation activities defined in <**5.2.1 security target specification evaluation**> and there are no additional evaluation methods and evaluation activities

2.3. Rationale for PP Declaration of compliance

TOE of this security target specification is a database encryption product that complies with the 'National Database Encryption Protection Profile V3.0' in a strict manner. In addition, there are no other protection profiles synthesized by the security target specification.

Classification		PP	ST	Rationale
TOE Type		Database Encryption	Database Encryption	Same as PP
Security Function Requirement(SFR)	Assets	Database managed by DBMS in an organization's operating environment	Database managed by DBMS in an organization's operating environment	Same as PP
		TOE itself and important data related to TOE operations (e.g., TSF data)	TOE itself and important data related to TOE operations (e.g., TSF data)	Same as PP
	Threats	T.SESSION_HIJACK	T.SESSION_HIJACK	Same as PP
		T.RETRY_AUTH_ATTEMPT	T.RETRY_AUTH_ATTEMPT	Same as PP
		T.IMPERSONATION	T.IMPERSONATION	Same as PP
		T.REPLAY	T.REPLAY	Same as PP
		T.WEAK_PASSWORD	T.WEAK_PASSWORD	Same as PP
	Organization Security	P.AUDIT	P.AUDIT	Same as PP
		P.SECURE_OPERATION	P.SECURE_OPERATION	Same as PP
		P.CRYPTO_STRENGTH	P.CRYPTO_STRENGTH	Same as PP
	Assumption	A.PHYSICAL_CONTROL	A.PHYSICAL_CONTROL	Same as PP
		A.TRUSTED_ADMIN	A.TRUSTED_ADMIN	Same as PP
		A.SECURE_DEVELOPMENT	A.SECURE_DEVELOPMENT	Same as PP
		A.OPERATION_SYSTEM_REINFORCEMENT	A.OPERATION_SYSTEM_REINFORCEMENT	Same as PP

Security Purpose	OE.LOG_BACKUP	OE.LOG_BACKUP	Same as PP
	OE.PHYSICAL_CONTROL	OE.PHYSICAL_CONTROL	Same as PP
	OE.TRUSTED_ADMIN	OE.TRUSTED_ADMIN	Same as PP
	OE.SECURE_DEVELOPMENT	OE.SECURE_DEVELOPMENT	Same as PP
	OE.OPERATION_SYSTEM_RE-INFORCEMENT	OE.OPERATION_SYSTEM_RE-INFORCEMENT	Same as PP
	-	OE.MANUAL_RECOVERY	More restrictive than PP. PP did not have a security purpose for manual recovery of TOE in the event of a failure or accident, but this ST added a security purpose for the operating environment to perform manual recovery in compliance with the instructions and procedures provided for normal recovery of TOE
	-	OE.DBMS	More restrictive than PP. Clearly define the safe storage and protection of audit and TSF data generated by TOE to strengthen association with existing security requirements and add security objectives to enhance data protection
	-	OE.TIME_STAMP	More restrictive than PP. To ensure the reliability of security-related events in the TOE operating environment and enhance the integrity and traceability of audit data, clearly define the use of reliable timestamps to strengthen association with existing security requirements and add security objectives
Security Function Requirement	FAU_ARP.1	FAU_ARP.1	Same as PP
	FAU_GEN.1	FAU_GEN.1	Same as PP
	FAU_SAA.1	FAU_SAA.1	Same as PP
	FAU_SAR.1	FAU_SAR.1	Same as PP
	FAU_SAR.3	FAU_SAR.3	Same as PP
	FAU_STG.1	FAU_STG.1	Same as PP

	FAU_STG.2	FAU_STG.2	Same as PP
	FAU_STG.4	FAU_STG.4	Same as PP
	FAU_STG.5	FAU_STG.5	Same as PP
	FCS_CKM.1(1)	FCS_CKM.1(1)	Same as PP
	FCS_CKM.1(2)	FCS_CKM.1(2)	Same as PP
	FCS_CKM.2	FCS_CKM.2	Same as PP
	FCS_CKM.5	FCS_CKM.5	Same as PP
	FCS_CKM.6	FCS_CKM.6	Same as PP
	FCS_COP.1(1)	FCS_COP.1(1)	Same as PP
	FCS_COP.1(2)	FCS_COP.1(2)	Same as PP
	FCS_RBG.1	FCS_RBG.1	Same as PP
	FCS_RBG.2	-	As a conditional mandatory SFR, it is not implemented as mandatory in TOE.
	FCS_RBG.3	-	As a conditional mandatory SFR, it is not implemented as mandatory in TOE.
	FCS_RBG.4	FCS_RBG.4	Same as PP
	FCS_RBG.5	FCS_RBG.5	Same as PP
	FDP_UDE.1	FDP_UDE.1	Same as PP
	FDP_RIP.1	FDP_RIP.1	Same as PP
	FIA_AFL.1	FIA_AFL.1	Same as PP
	FIA_IMA.1	FIA_IMA.1	Same as PP
	FIA_SOS.1	FIA_SOS.1	Same as PP
	FIA_UAU.1	FIA_UAU.1	Same as PP
	FIA_UAU.4	FIA_UAU.4	Same as PP
	FIA_UAU.5	-	As a conditional mandatory SFR, it is not implemented as mandatory in TOE.
	FIA_UAU.7	FIA_UAU.7	Same as PP
	FIA_UID.1	FIA_UID.1	Same as PP
	FMT_MOF.1	FMT_MOF.1	Same as PP
	FMT_MTD.1	FMT_MTD.1	Same as PP
	FMT_PWD.1	FMT_PWD.1	Same as PP
	FMT_SMF.1	FMT_SMF.1	Same as PP
	FMT_SMR.1	FMT_SMR.1	Same as PP
	FPT_FLS.1	FPT_FLS.1	Same as PP
	FPT_ITT.1	FPT_ITT.1	Same as PP
	FPT_LEE.1	-	As a conditional mandatory SFR, it is not implemented as mandatory in TOE.
	FPT_PST.1	FPT_PST.1	Same as PP

	FPT_RCV.1	-	As a conditional mandatory SFR, it is not implemented as mandatory in TOE.
	FPT_RCV.2	-	As a conditional mandatory SFR, it is not implemented as mandatory in TOE.
	FPT_STM.1	-	As a Optional SFR, it is not implemented as mandatory in TOE.
	FPT_TST.1	FPT_TST.1	Same as PP
	FPT_TUD.1	-	As a conditional mandatory SFR, it is not implemented as mandatory in TOE.
	FTA_MCS.2	FTA_MCS.2	Same as PP
	FTA_SSL.1	-	As a conditional mandatory SFR, it is not implemented as mandatory in TOE.
	FTA_SSL.3	FTA_SSL.3	Same as PP
	FTA_TSE.1(1)	FTA_TSE.1(1)	Same as PP
	FTA_TSE.1(2)	-	As a conditional mandatory SFR, it is not implemented as mandatory in TOE.
	FTP_ITC.1	-	As a conditional mandatory SFR, it is not implemented as mandatory in TOE.
	FTP_TRP.1	-	As a conditional mandatory SFR, it is not implemented as mandatory in TOE.
Assurance Requirement	AGD_OPE.1	AGD_OPE.1	Same as PP
	AGD_PRE.1	AGD_PRE.1	Same as PP
	ALC_CMC.1	ALC_CMC.1	Same as PP
	ALC_CMS.1	ALC_CMS.1	Same as PP
	ASE_CCL.1	ASE_CCL.1	Same as PP
	ASE_ECD.1	ASE_ECD.1	Same as PP
	ASE_INT.1	ASE_INT.1	Same as PP
	ASE_OBJ.1	ASE_OBJ.1	Same as PP
	ASE_REQ.1	ASE_REQ.1	Same as PP
	ASE_TSS.1	ASE_TSS.1	Same as PP
	ATE_FUN.1	ATE_FUN.1	Same as PP
	ATE_IND.1	ATE_IND.1	Same as PP
	AVA_VAN.1	AVA_VAN.1	Same as PP

[Table 18] Theoretical basis for compliance

3. Security Problems

3.1 Definition of Security Issue

The security issue definition defines threats, security policies, and assumptions of the organization that TOE and the TOE operating environment are intended to address.

3.1.1 Assets

The basic assets protected by database encryption are as follows.

- Database managed by DBMS in an organization's operating environment
- TOE itself and important data related to TOE operations (e.g., TSF data)

3.1.2 Threats

Threat sources are IT entities and users who cause harm to assets to be protected by unauthorized access or abnormal methods, and can cause various threats as follows. At this time, the threat to TOE has a basic level of expertise, resources, and motivation.

3.1.2.1 Unauthorized Access

T.SESSION_HIJACK

The threat source may steal user rights by accessing a user screen left logged in or using a user session that has not ended the session in the logout state.

T.RETRY_AUTH_ATTEMPT

After successful authentication using the information obtained by continuously attempting authentication, the threat source may access the TOE by disguising it as an authorized user.

T.IMPERSONATION

The threat sources may access the TOE by disguising it as an authorized user, a TOE component, and the like.

T.REPLAY

Threat sources can identify and copy authentication information and reuse it to access TOE.

T.WEAK_PASSWORD

The threat source can access the TOE after obtaining a password that is insufficiently managed, such as using the default value of the password, disguising it as an authorized user, and when a low-level password rule is applied, it can access the TOE by disguising it as an authorized user.

3.1.2.2 Information Leakage

T. UNAUTHORIZED_INFO_LEAK

The threat sources may leak user important information stored in the database in an unauthorized manner.

T.STORED_DATA_LEAKAGE
Threat sources may leak critical data (e.g., encryption keys, TOE settings, etc.) stored inside the TOE or in an external entity (e.g., DBMS) that interacts with the TOE in an unauthorized manner.

T.TRANSMISSION_DATA_DAMAGE
Threat sources may expose and change transmission data between components of TOE and with external IT entities in an unauthorized manner.

T.WEAK_CRYPTO_PROTOCOLS
Threat sources may infer encryption key information or find out the content of the communication cipher text by analyzing traffic using weak encryption communication protocols or low encryption strength.

3.1.2.3 TOE Function damages

T.TSF_COMPROMISE
The threat source may damage the TSF through unauthorized access, etc., causing malfunction of the TOE function or disabling the TOE function.

3.1.3 Security Policies of Organization

P.AUDIT
Security-related events should be recorded and maintained to track responsibility for security-related actions, and recorded data should be reviewed. In addition, the available space on the disk used to store audit data should be checked regularly to prevent audit data from being lost and Unauthorized changes and deletions to stored audit data should be protected.

P.SECURE_OPERATION
The administrator shall provide management means to ensure that the administrator's security policy to comply with the organization and operate correctly according to the organization.

P.CRYPTO_STRENGTH
Organizations should apply encryption measures for storage and transmission sections of important data, such as passwords for user authentication, and use secure encryption algorithms.

3.1.4 Assumptions

It is assumed that the following conditions exist in the TOE operating environment accommodating this protection profile.
A.PHYSICAL_CONTROL
The place where the TOE is installed and operated shall be equipped with access control and protection

facilities so that only authorized managers can access it.

A.TRUSTED_ADMIN

Authorized managers of TOE are not malicious, have been properly trained in TOE management functions and perform their duties accurately according to manager guidelines.

A.SECURE_DEVELOPMENT

Developers who use TOE to link the encryption function to the application or DBMS must comply with the requirements of the manual provided with the TOE to ensure that the security function of the TOE is safely applied.

A.OPERATION_SYSTEM_REINFORCEMENT

It shall ensure the reliability and stability of the operating system by performing reinforcement work on the latest vulnerabilities of the operating system in which the TOE is installed and operated.

3.2. Security objectives

The security objectives of the following operating environment should be addressed by technical/procedural means supported by the operating environment so that TOE can accurately provide security functionality.

3.2.1. Security objectives for the operational environment

OE.PHYSICAL_CONTROL

The place where TOE components are installed and operated shall be equipped with access control and protection facilities so that only authorized administrator can access.

OE.TRUSTED_ADMIN

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

OE.SECURE_DEVELOPMENT

Developers who use TOE to link the encryption function to the application or DBMS must comply with the requirements of the manual provided with the TOE to ensure that the security function of the TOE is safely applied.

OE.LOG_BACKUP

TOE's authorized manager shall periodically check the free space of the audit data storage in preparation for the loss of audit records and back up audit records (external log servers, separate storage devices, etc.) so that audit records are not exhausted.

OE.OPERATION_SYSTEM_RE-INFORCEMENT

TOE's authorized manager shall ensure the reliability and safety of the operating system by performing reinforcement work on the latest vulnerabilities of the operating system in which the TOE is installed and operated.

OE.MANUAL_RECOVERY
TOE's authorized manager shall perform manual recovery operations in compliance with the instructions and procedures provided to restore the normal state of the TOE in the event of a failure or accident.

OE.DBMS
DBMS shall safely store and protect audit data and TSF data generated by TOE.

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

3.2.2. Theoretical basis for Security Purposes

3.2.2.1. Theoretical basis for security purposes to operational environment

	OE.Log Backup	OE.Physical Security	OE.Trusted Manager	OE.Safe Development	OE.Operating System Reinforcement	OE.Manual Recovery	OE.DBMS	OE.Timestamp
P.Audit	O						O	O
P.Safe Operationial			O			O		
A.PhysicalSecurity		O						
A.Trusted Manager	O		O					
A.Safe Development				O				
A.Operating System Reinforcement					O			

[Table 19] Defining security issues and responding to the operational environment for security purposes

P.Audit OE.Log Backup
P.Audit shall be fulfilled by OE.Log Backup
OE.Logbackup not only performs TOE functions, but also regular audit data storage space checks by managers, and It shall perform regular log backups or log transmission to external log servers to prevent

log records from being lost.

P.Safe Operation OE.Trusted Manager
P.Safe Operation shall be fulfilled by OE.Trusted Manager.
OE. A trusted manager ensures that the manager operates the TOE accurately in accordance with the organization's security policy and operational manual.

A.Physical Security OE. Physical Security
A.Physical security is supported by OE.Physical security.
OE.Physical security places management servers in places with protective facilities and controls access so that only authorized managers can access them.

A.Trusted Manager OE.Trusted Manager, OE.LogBackup
A.Trusted Manager are supported by OE.Trusted Manager, OE.log backups.
OE.Trusted Managers are innocuous and ensure that they are properly trained in TOE management functions and perform their duties accurately in accordance with manager guidelines.
OE.LogBackup ensures that authorized managers periodically check the free space in the audit data storage in preparation for the loss of audit records and back up audit records (external log servers, additional store devices, etc.) so that audit records are not exhausted.

A.Safe Development OE.Safe Development
A.Safe development is supported by OE.Safe development.
OE.Safe development ensures that TOE's security features are safely applied by developers who use TOE to link encryption functions to the application or DBMS by complying with the requirements of the documentation provided with the TOE.

A.O/S Reinforcement OE. O/S Reinforcement
A.Operating System Reinforcement is supported by OE.O/S Reinforcement
OE. Operating system reinforcement ensures reliability and safety of the operating system by performing reinforcement work on the latest vulnerabilities of the operating system in which TOE is installed and operated.

P.Safe Operation OE.Manual Recovery
P.Safe operation is supported by OE. Manual Recovery.
OE.Manual recovery ensures that administrators can recover the system through appropriate procedures even in situations where TOE is not operating normally due to failure or error. Through this, important security functions are maintained continuously, and P.safe operation of the organization's security policy can be implemented through safe recovery.

P.Audit OE.Database

P.Audit shall be fulfilled by the OE. database.

The OE. database ensures that the audit data generated by the TOE is stored securely in a reliable storage space, and is necessary to conduct security policy P.audit of the organization by preventing the loss and tampering of audit data.

P. Audit OE.Timestamp

P.Audit ensures that TOE accurately records security-related events using reliable timestamps provided by the TOE operating environment, so it is necessary to conduct an organization's security policy P.audit.

P.Safe Operation P. Audit

an administrator operates the TOE accurately in accordance with the organization's security policies and operational manuals, and such administrative actions are recorded in audit logs. The generated audit data is securely stored in a trusted repository, and security events are precisely timestamped using a reliable timestamp mechanism.

4. Extended Components Definition

Since this Security Target Specification complies with the Common Evaluation Criteria 2 and 3, there are no extended components.

If there are any extension components that have been complied with by PP, they are strictly compliant, so they are all applied.

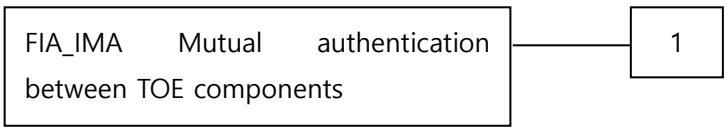
4.1. Identification & authentication (FIA)

4.1.1. Mutual authentication between TOE components

Family Behavior

TOE Internal Mutual Authentication (FIA_IMA, TOE IMA) family requires that mutual authentication functions between TOE components be provided in the process of user identification and authentication.

Component Hierarchical Relation and Description



FIA_IMA.1 Mutual authentication between TOE components requires that mutual authentication between TOE components be provided in the process of user identification and authentication.

Management : FIA_IMA.1

There are no management activities foreseen.

Audit : FIA_IMA.1

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

a) Minimal: Success and failure of mutual authentication

4.1.1.1. FIA_IMA.1 TOE Mutual authentication between TOE components

Hierarchical to No other components

Dependencies No dependencies

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

4.2. User data protection (FDP)

4.2.1. User Data encryption

Family Behavior

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

Component Hierarchical Relation and Description

FDP_UDE.1 User Data Encryption requires that confidentiality of user data is guaranteed.

Management : FDP_UDE.1

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

a) Management of user data encryption/decryption rules

Audit : FDP_UDE.1

If the FAU_GEN security audit data generation family is included in PPs, PP-modules, functional packages, or STs, it is recommended to audit and record the following actions

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

4.2.1.1. FDP_UDE.1 User Data encryptioin

Hierarchical to	No other components
Dependencies	FCS_COP.1 Cryptographic operation

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

4.3. Security Management(FMT)

4.3.1. ID and Password

Family Behavior

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

Component Hierarchical Relation and Description

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

Management : FMT_PWD.1
The following actions could be considered for the management functions in FMT:
a) Management of ID and password configuration rules

Audit : FMT_PWD.1
The following actions are recommended to record if FAU_GEN Security audit data generation is included in PPs, PP-modules, functional packages, or STs:
a) Minimal: All changes of the password

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

FMT_PWD.1.2 The TSF shall restrict the ability to manage the ID of [assignment: *list of functions*] to [assignment: *the authorized identified roles*].
1.[assignment : *ID combination rules and/or length*]
2.[assignment : *other management such as management of special characters unusable for ID, etc.*]

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

4.4. Protection of the TSF(FPT)

4.4.1. Protection of stored TSF data

Family Behavior

The family defines rules for protecting TSF data stored in TSF-controlled storage from unauthorized changes or exposure

Component Hierarchical Relation and Description

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

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

Audit : FPT_PST.1
There are no auditable events foreseen.

4.4.1.1. FPT_PST.1 Basic protection of stored TSF data

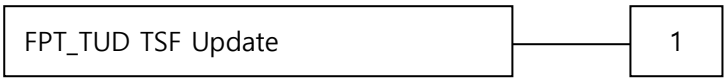
Hierarchical to	No other components
Dependencies	No dependencies

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

4.4.2. TSF Update

Family Behavior
The family defines TOE firmware/software update requirements.

Component Hierarchical Relation and Description



FPT_TUD.1 TSF security patch updates require to ensure trusted updates of TOE firmware/software, including the ability to validate the update file before installing the update.

Management : FPT_TUD.1
The following management functions may be considered in FMT.
Managing Update File Verification Mechanism

Audit : FPT_TUD.1

FAU_GEN If the security audit data generation family is included in a PP, PP-module, functional package, or ST, it is recommended to audit and record the following actions.

a) a) Minimum: Update file integrity verification results (success, failure)

5. Security Requirements

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

5.1. Security Functional Requirements

The Security Function Requirements defined in this ST are expressed by selecting the relevant security function components from CC Part 2 to satisfy the security objectives identified in Chapter 4.
The following table provides a summary of the security function components used in this ST.

Security Function Class	Security Functional Component		Notes
Security Audit (FAU)	FAU_ARP.1	Security alarms	Mandatory SFR
	FAU_GEN.1	Audit data generation	Mandatory SFR
	FAU_SAA.1	Potential violation analysis	Mandatory SFR
	FAU_SAR.1	Audit review	Mandatory SFR
	FAU_SAR.3	Selective audit	Mandatory SFR
	FAU_STG.1	Spot to store audit data	Mandatory SFR
	FAU_STG.2	Audit data storage protection	Conditional Mandatory SFR
	FAU_STG.4	Response activity to forecast loss of audit data	Conditional Mandatory SFR
	FAU_STG.5	Prevention of audit data loss	Conditional Mandatory SFR
Cryptographic Support (FCS)	FCS_CKM.1(1)	Cryptographic key generation (User data encryption)	Mandatory SFR
	FCS_CKM.1(2)	Cryptographic key generation (TSF Data Encryption)	Mandatory SFR
	FCS_CKM.2	Cryptographic key distribution	Optional SFR
	FCS_CKM.5	Cryptographic key derivation	Conditional Mandatory SFR
	FCS_CKM.6	Cryptographic key destruction time and incident	Mandatory SFR
	FCS_COP.1(1)	Cryptographic operation(User Data Encryption)	Mandatory SFR
	FCS_COP.1(2)	Cryptographic operation(TSF Data Encryption)	Mandatory SFR
	FCS_RBG.1	Random bit	Mandatory SFR

	FCS_RBG.4	Random bit(internal seeding – multiple source)	Conditional Mandatory SFR
	FCS_RBG.5	Random bit(entropy source combination)	Conditional Mandatory SFR
User Data Protection (FDP)	FDP_UDE.1	User data encryption(extended)	Mandatory SFR
	FDP_RIP.1	Partial remaining information protection	Mandatory SFR
Identification and Authentication(FIA)	FIA_AFL.1	Authentication failure handling	Mandatory SFR
	FIA_IMA.1	TOE mutual authentication among component(extended)	Mandatory SFR
	FIA_SOS.1	Verification of secrets	Mandatory SFR
	FIA_UAU.1	Authentication	Mandatory SFR
	FIA_UAU.4	Anti-reuse authentication mechanism	Mandatory SFR
	FIA_UAU.7	Authentication feedback protection	Mandatory SFR
	FIA_UID.1	Identification	Mandatory SFR
Security Management (FMT)	FMT_MOF.1	Management of security function	Mandatory SFR
	FMT_MTD.1	Management of TSF data	Mandatory SFR
	FMT_PWD.1	Management of ID and Password(extended)	Mandatory SFR
	FMT_SMF.1	Specification of management functions	Mandatory SFR
	FMT_SMR.1	Security roles	Mandatory SFR
TSF Protection (FPT)	FPT_FLS.1	Stay safe in case of failure	Mandatory SFR
	FPT_ITT.1	Basic protection of internal transport TSF	Mandatory SFR
	FPT_PST.1	Basic protection of stored TSF data(extended)	Mandatory SFR
	FPT_TST.1	TSF self test	Mandatory SFR
TOE Access (FTA)	FTA_MCS.2	Limitation the number of concurrent sessions by user attribute	Mandatory SFR
	FTA_SSL.3	Session termination by TSF	Conditional Mandatory SFR
	FTA_TSE.1(1)	TOE session setting	Mandatory SFR

[Table 20] Security Function Requirements

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 perform [email notification to an authorized administrator] upon

detection of a potential security violation.

5.1.1.2. FAU_GEN.1 Audit data generation

Hierarchical to No other components

Dependencies FPT_STM.1 Reliable time stamps

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

a) Start-up and shutdown of the audit functions.

b) All auditable events for the *not specified* level of audit, and

c) [Refer to "auditable event" in [Table 21] Auditable Event

Security Function Component	Auditable Event	Additional Audit Record
FAU_STG.5	Response actions and results (success, failure) when audit storage fails	
FCS_CKM.1	Failure to generate encryption key	
FCS_COP.1	Failure of encryption operation(including encryption operation type)	
FDP_UDE.1	Success and failure of user data encryption/decryption	
FIA_AFL.1	Response actions and results (success, failure) when user authentication attempt limit is reached	
FIA_UAU.1	User login success/failure	
FIA_UAU.4	Authentication failure due to detection of authentication information reuse attempt	
FMT_MOF.1	All security management actions for [Assignment: Function List] specified in FMT_MOF.1.1	Changed TSF data value
FMT_MTD.1	User registration, change, deletion	Changed TSF data value
	All changes to passwords	
	Security management actions for 'TSF data list' specified in FMT_MTD.1.1	
	TOE agent registration status change	
FMT_PWD.1	Default account (ID), password change	
FPT_TST.1	TOE server self-test performance and results (success, failure)	Failed security function
	TOE component integrity verification performance and results (success, failure)	Component which failed integrity check
FTA_MCS.2	New session rejection based on limit on number of concurrent sessions	
	Response actions when detecting duplicate logins of the same account	
	Duplicate access blocking and Result (success/failure)	
FTA_SSL.3	Perform session termination of user and result (success/failure)	

FTA_TSE.1	Block IP of terminal access for management	
Etc	Success and failure of user logout	

[Table 21] Auditable Event

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:
a) Date and time of the event, type of event, subject identity (if possible) and the outcome (success of failure) of the event
b) For each audit event type, ["Additional audit record" in [Table 21]] based on the audit target event definition of the functional component included in the security target specification

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) [Failure of self-test, failure of integrity verification, audit of FPT_TST.1] known to indicate potential security violations.
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 administrators] with the ability to read [all audit data] from the audit record**.**.
FAU_SAR.1.2 The TSF shall provide audit records suitable for interpretation by authorized administrators.

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 [the following conditions (AND)] of audit data based on [the following conditions (AND) or Ordering Method]**.**

Audit Data Type	Condition (AND)	Selection or Ordering Method
Administrator Log	Date and Time	Search results are sorted in descending order based on the selected search criteria.
	Result Message	

[Table 22] Audit Data Type

5.1.1.6. FAU_STG.1 Where to store Audit Data

Hierarchical to No other components

Dependencies FAU_GEN.1 Generating Audit Data

FTP_ITC.1 Safe Channel among TSF

FAU_STG.1.1 TSF shall be able to *[store in Local DBMS]* generated audit data.

5.2.1.7. FAU_STG.2 Audit data storage protection

Hierarchical to No other components

Dependencies FAU_GEN.1 Generating Audit Data

FAU_STG.2.1 TSF shall protect audit records stored within audit records from unauthorized deletion.

FAU_STG.2.1 TSF shall *prevent* unauthorized changes to audit records stored within audit records.

5.2.1.8. FAU_STG.4 Response to forecast loss of audit data

Hierarchical to No other components

Dependencies FAU_STG.1 Audit data storage protection

FAU_STG.4.1 TSF shall take [Notify Authorized Manager, None] if the audit trail exceeds [90% of the total row of DB table].

5.2.1.9. FAU_STG.5 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.5.1 If the audit trail is saturated, TSF shall perform *to overwrite the oldest audit record* and [to notify the authorized administrator].

5.1.2. . Cryptographic Support (FCS)

5.1.2.1. FCS_CKM.1(1) Cryptographic key generation (User Data Encryption)

Hierarchical to No other components

Dependencies [FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation]

FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1 TSF shall generate **Data Encryption Key(DEK)** according to the specified encryption key algorithm [HASH_DRBG (SHA 256)] and the specified encryption key length [256 Bit] that conforms to the following [TTAK.KO-12.0331]

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.1.1 TSF shall generate encryption keys according to the specified encryption key generation algorithm [Key Generation Algorithm in [Table 23]] and the specified encryption key length [Cryptographic Key Length in [Table 23]] in accordance with the following [Standard [Table 23]].

Standard	Key Generation Algorithm	Cryptographic Key Length	Encryption Key Usage
TTAK.KO-12.0331	HASH_DRBG (SHA 256)	256	Encryption and decryption of configuration Encryption and decryption of security policy file, transport data, and DB encryption key, Session key distribution, Encryption and decryption of backup file
TTAK.KO-12.0334-Part1/2 (2018)	PBKDF	256	Encryption and decryption Master key

[Table 23] TSF Data Encryption Key Generation Standards and Algorithms

Note for application: When generating a key (KEK) for key encryption by deriving it from a password, the pseudo-random function uses HMAC-SHA2 and the iteration count is 100,000.

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 without security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.2.1 TSF shall distribute encryption keys in accordance with the specified encryption key

distribution method [ECDHE-based key distribution method] that conforms to the following [none]

5.1.2.4. FCS_CKM.5 Cryptographic Inducement

Hierarchical to: No other components

Dependencies: [FCS_CKM.2 Cryptographic Key distribution or
FCS_COP.1 Cryptographic operation
FCS_CKM.6 Cryptographic Key destruction time and events]

FCS_CKM.5.1 TSF must induce the encryption key [key encryption key (KEK)] from the [password input] according to the specified key derivation algorithm [PBKDF] and the specified encryption key length [256 bits] that conform to the following [TTAK.KO-12.0334-Part1/2 (2018)].

5.1.2.5. FCS_CKM.6 Cryptographic Key Destruction Time and Events

Hierarchical to: No other components

Dependencies: [FDP_ITC.1 Inflow of user data without security properties or
FDP_ITC.2 Inflow of user data with security properties or
FCS_CKM.1 Cryptographic key generation or
FCS_CKM.5 Cryptographic key inducement]

FCS_CKM.6.1 TSF should destroy [key for key encryption (KEK), KEK derivation password, user data encryption key, TSF data encryption key, certificate private key] *when no longer required*.

FCS_CKM.6.2 TSF shall destroy the encryption key and key material specified in FCKS_CKM6.1 in accordance with the specified encryption key destruction method [overwrite 3 times with '0'] that meets the following [none].

5.1.2.6. FCS_COP.1(1) Cryptographic Computing (User data encryption)

Hierarchical to: No other components

Dependencies: [FDP_ITC.1 Inflow of user data without security properties or
FDP_ITC.2 Inflow of user data with security properties or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1 TSF shall perform [Computing List in [Table 24]] according to [Cryptographic Algorithm in [Table 24]] and [Cryptographic Key Length in [Table 24]] which conforms to the following [Standard in [Table 24]].

Standard	Cryptographic Algorithm	Cryptographic Key Length	Operation Mode	Computing List

KS X 1213-1	ARIA	256	CBC	En/Decrypt user data stored in DB
ISO/IEC 10118-3	SHA-256, SHA-512	N/A	N/A	Encrypt user data stored in DB

[Table 24] Cryptographic Computing Standard and algorithm

5.1.2.7. FCS_COP.1(2) Cryptographic Computing (TSF data encryption)

Hierarchical to: No other components

Dependencies: [FDP_ITC.1 Inflow of user data without security properties or
FDP_ITC.2 Inflow of user data with security properties or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1 TSF shall perform [Computing List in [Table 25]] according to [Cryptographic Algorithm in [Table 25]] and [Cryptographic Key Length in [Table 25]] which conforms to the following [Standard in [Table 25]].

Standard	Cryptographic Algorithm	Cryptographic Key Length	Operation Mode	Computing List
KS X 1213-1	ARIA	256	CBC	En/Decryption of security policy file, transport data, and DB encryption key Encryption and decryption of backup file
ISO/IEC 14888-2	RSA-PSS	2048	N/A	Mutual Authentication, Module and Configuration Integrity verification
ISO/IEC 10118-3	SHA-256	N/A	N/A	Policy File Intergrity Verification, Administrator password encryption

[Table 25] List of Cryptographic algorithm

5.1.2.8. FCS_RBG.1 Random Bit Generation(RBG)

Hierarchical to: No other components

Dependencies: [FCS_RBG.2 Random bit generation(external seeding) or
FCS_RBG.3 Random bit generation(internal seeding – sole source)]
FPT_FLS.1 Stay safe in case of failure
FPT_TST.1 TSF self test

FCS_RBG.1.1 After initialization, TSF shall perform deterministic random bit generation services using [HASH-DRBG (SHA 256)] in accordance with [TTAK.KO-12.0331-Part 2].

FCS_RBG.1.2 TSF shall use TSF *entropy source [/dev/urandom, Jitter]* for initialization and seeding.

FCS_RBG.1.3 TSF shall update DRBG status by *re-seeding* using TSF *entropy source [/dev/urandom, Jitter]* in accordance with [TTAK.KO-12.0331-Part2] in the following situations.
The following situations [initialization, reaching the update cycle (100,000)]

5.1.2.9. FCS_RBG.4 Random Bit Generation(Internal Seeding – Multi Source)

Hierarchical to No other components

Dependencies FCS_RBG.1 Random bit generation(RBG)
FCS_RBG.5 Random bit generation(Entropy Source Combination)

FCS_RBG.4.1 TSF shall be able to seed DRBG using *[two] TSF software-based entropy source*

5.1.2.10. FCS_RBG.5 Random Bit Generation(Entropy Source Combination)

Hierarchical to No other components

Dependencies FCS_RBG.1 Random bit generation(RBG)
[FCS_RBG.2 Random bit generation(external seeding) of
FCS_RBG.3 Random bit generation(internal seeding – sole source)
FCS_RBG.4 Random bit generation(internal seeding – multi source)]

FCS_RBG.5.1 TSF must perform [connections] of the *output from the TSF entropy source(s)* that generate at least [384 bits] minimum entropy to generate an entropy input for the derivative function defined in [TTAK.KO-12.0331-Part2] so that it is at least [512] bits minimum entropy.

5.1.3. User Data Protection(FDP)

5.1.3.1. FDP_UDE.1 User data encryption(extended)

Hierarchical to No other components

Dependencies FCS_COP.1 Cryptographic Computing

FDP_UDE.1.1 TSF should provide TOE users with the ability to encrypt and decrypt user data according to the specified [column encryption/decryption method, [none]].

5.1.3.2. FDP_RIP.1 Partial Remaining Information Protection

Hierarchical to No other components

Dependencies No dependencies

FDP_RIP.1.1 TSF shall ensure that all previous information of the resource is not available when *allocating and retrieving the resource to 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 [*5*] unsuccessful authentication attempts occur related to [*Administrator Authentication Attempts*].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been *met,* the TSF shall [lock the screen for 5 minutes and notify the administrator of mail].

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

Hierarchical to No other components

Dependencies No dependencies

FIA_IMA.1.1 TSF shall perform mutual authentication using [Self-authentication protocol] in accordance with [None] between [PrivacyConsole and PrivacyKMS, PrivacyKMS and PrivacyEOC].

5.1.4.3. FIA_SOS.1 Verification of secrets

Hierarchical to No other components

Dependencies No dependencies

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet [the following defined quality metric].

[Password combination rules]

- Digit : more than 10 ~ less then 40 character
- numbers, uppercase letters (English), lowercase letters (English), special character include at least on each
- Prohibit setting for User Account(ID) and the same Password
- Prohibit enter the same character or number repeatedly
- Prohibit sequential input of consecutive characters or numbers on the keyboard,
- Prohibit reuse previous password.
- Number(10) : 0~9,
- Uppercase Letters(26) : A~Z,
- Lowercase Letters(26) : a~z,
- Special Character(32) : `~!@#$%^&*()-_+=[]{}₩|;:'",.<>/?

5.1.4.4. FIA_UAU.1 Authentication

Hierarchical to No other components

Dependencies FIA_UID.1 Identification

FIA_UAU.1.1 The TSF shall allow [certificate generation] to be performed on behalf of the user before the user is authenticated.

FIA_UAU.1.2 TSF shall successfully authenticate the user before allowing all other actions mediated by TSF on behalf of the user other than those specified in FIA_UAU.1.1.

5.1.4.5. FIA_UAU.4 Anti-reuse Authentication Mechanism

Hierarchical to No other components

Dependencies No dependencies

FIA_UAU.4.1 The TSF shall prevent the reuse of authentication data related to [Administrator Authentication].

5.1.4.6. FIA_UAU.7 Authentication Feedback Protection

Hierarchical to No other components

Dependencies FIA_UAU.1 Authentication

FIA_UAU.7.1 The TSF shall provide only ['*', a message that cannot infer the reason for failure in the event of authentication failure] to the user while the authentication is in progress

5.1.4.7. FIA_UID.1 Identification

Hierarchical to No other components

Dependencies No dependencies

FIA_UID.1.1 The TSF should allow [certificate generation] to be performed on behalf of the user before identifying the user.

FIA_UID.1.2 TSF shall successfully identify each user before allowing all other actions mediated by TSF on behalf of the user other than those specified in FIA_UID.1.1.

5.1.5. Security Management(FMT)

5.1.5.1. FMT_MOF.1 Security Function Management

Hierarchical to No other components

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

FMT_MOF.1.1 TSF shall restrict the ability to ***management Behavior*** [Security Function of [Table 26]] to [Authorized Administrator].

Administrator Type	Management Type	Security Function	Management Behavior Capability			
			Determine the behavior	Stop	Start	Changing the behavior
Authorized Administrator	Encryption Key	Generating En/Decryption Key	○	–	○	–
	Security Policy	Encryption Target Type	○	○	○	○
		Type of encryption algorithm	○	○	○	○
		User Data Integrity Check Feature	○	○	○	○
		Double Encryption	○	○	○	○
		Encryption Pattern	○	○	○	○
	Access Rights	User Access	○	○	○	○
	Configuration	User Access Rights Permit / Deny Policy	○	○	–	○
		Key Policy backup	○	–	○	–
		Administrator IP	–	○	○	○
		Mail Server	–	○	○	○
	Audit Log	Selecting of audit targets(cipher text)	○	○	○	○
		Creating a success log	○	○	○	○

[Table 26] Security Function Management of Authorized Administrator

5.1.5.2. FMT_MTD.1 TSF Data Management

Hierarchical to No other components

Dependencies FMT_SMF.1 Specification of management functions

FMT_SMR.1 Security roles

FMT_MTD.1.1 TSF shall restrict the ability to ***manage*** [Data Type of [Table 27]] to [Authorized Administrator].

Administrator Type	Management Type	Data Type	Management Behavior Capability				
			Changing Default	Inquiry	Change	Creation	Deletion
Authorized Administrator	Key	User Data En/Decryption Key	–	○	–	○	○
		Master Key	–	–	○	○	–
	User	DB User	–	○	○	○	○
	Security Policy	User Data Cryptographic Policy	○	○	○	○	○
	Access Authority	Access Time	○	○	–	○	○
		Access User	○	○	–	○	○
		Access IP	–	○	○	○	○
		Access Application	–	○	○	○	○
	Configuration	Key Policy backup	–	○	–	○	–
		Admin IP	–	○	○	○	○
		Mail Server	–	○	○	○	○
	Audit Log	Admin Log	–	○	–	–	–
		Encryption Log	–	○	–	–	–
	Authentication Information	Password	–	–	○	○	–

[Table 27] List of TSF Data Management Capability

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

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

FMT_PWD.1.3 The TSF shall provide the ability to set the ID and password during the installation process.

5.1.5.4. FMT_SMF.1 Specification of management functions

Hierarchical to No other components
Dependencies No dependencies

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: [list of management functions to be provided by the TSF]
- Management functions of the TSF: Management functions specified in FMT_MOF.1
- Management of TSF data: Management functions specified in FMT_MTD.12
- Management of security role: Management functions specified in FMT_SMR.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 [Authentication Administrator].

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

5.1.6. TSF Protection(FPT)

5.1.6.1. FPT_FLS.1 Stay safe in case of a failure

Hierarchical to No other components
Dependencies No dependencies

FPT_FLS1.1 The TSF shall stay safe in the event of the following types of failures: [Noise source health test failed]

5.1.6.2. FPT_ITT.1 Basic internal TSF data transfer protection

Hierarchical No other components

Dependencies No dependencies

FPT_ITT.1.1 TSF shall protect TSF data from *exposure and alteration* when TSF data is transferred between separate parts of TOE

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

Hierarchical to No other components
Dependencies No dependencies

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

- Administrator authentication data
- Database access account information
- Encryption key
- TOE setting value (configuration, security policy settings, etc.)
- backup file

5.1.6.4. FPT_TST.1 1 TSF self test

Hierarchical to No other components
Dependencies No dependencies

FPT_TST.1.1 To demonstrate the correct operation of *TSF*, TSF shall run the following self-tests [Self-test List in Table 28] *upon start-up, periodically during regular operation, upon request of authorized administrators*.**.**

Condition	Self-test List
Start-up	Process status checks, cryptographic module self-test (self-test, health test), integrity verification
Periodically	Process status checks, cryptographic module self-test (self-test, health test), integrity verification
Request of authorized administrator	Integrity Verification

[Table 28] List of self-test executed by TSF

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

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

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 Identification

FTA_MCS.2.1 TSF shall limit the maximum number of concurrent sessions belonging to the same user according to the [Maximum number of users with the same privileges and concurrent sessions of the same user is limited to 1, none] rule.

FTA_MCS.2.2 TSF shall basically enforce [1] session limits per user.

5.1.7.2. FTA_SSL.3 Session termination by TSF

Hierarchical to No other components

Dependencies No dependencies

FTA_SSL.3.1 TSF shall terminate the interactive session after [10 minutes].

5.1.7.3. FTA_TSE.1(1) TOE session setting

Hierarchical to No other components

Dependencies No dependencies

FTA_TSE.1.1 TSF shall be able to deny the **administrator's management access session** setting based on [access IP, *[whether administrator accounts with the same privileges are active for administrative access sessions]*]

5.2. Assurance Requirements

The Assurance requirements of this security target specification consist of the warranty components of CC 5 parts, and the evaluation guarantee grade is EAL1+. The following table summarizes the warranty components.

Assurance Class	Assurance Component	
Security Target Evaluation	ASE_INT.1	ST introduction
	ASE_CCL.1	Conformance claims
	ASE_OBJ.1	Security objectives for the operational environment
	ASE_ECD.1	Extended components definition
	ASE_REQ.1	Stated security requirements

	ASE_TSS.1	TOE summary specification
Development	ADV_FSP.1	Basic functional specification
Guidance Documents	AGD_OPE.1	User operational guidance
	AGD_PRE.1	Preparative procedures
Life-cycle support	ALC_CMC.1	TOE Labeling
	ALC_CMS.1	TOE CM coverage
Tests	ATE_FUN.1	Functional testing
	ATE_IND.1	Independent testing : Function check
Vulnerability Assessment	AVA_VAN.1	Vulnerability survey

[Table 29] Assurance requirements

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 summaries the usage and major security features of the TOE.

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

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

ASE_INT.1.7C 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

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

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

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

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

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

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

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

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

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

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

Evaluator action elements

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

5.2.1.3. ASE_OBJ.1 Security objectives for the operational environment

Dependencies No dependencies

Developer action elements

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

Content and presentation elements

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

Evaluator action elements

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

5.2.1.4. ASE_ECD.1 Extended components definition

Dependencies No dependencies

Developer action elements

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

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

Content and presentation elements

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

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

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

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

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

Evaluator action elements

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

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

5.2.1.5. ASE_REQ.1 Stated security requirements

Dependencies ASE_ECD.1 Extended components definition

Developer action elements

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

ASE_REQ.2.2D 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 categorization of interface 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 User operational 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 privilege 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 operational user guidance shall be clear and reasonable.

5.2.3.2. AGD_PRE.1 Preparative procedures

Dependencies No dependencies

Developer action elements

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

Content and presentation elements

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

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

Evaluator action elements

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

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

5.2.4. Life-cycle Support

5.2.4.1. ALC_CMC.1 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 TOE shall label for sole reference.

Evaluator action elements

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

5.2.4.2. 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 followings: the TOE itself and the evaluation evidence required by the SARs.

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

Evaluator action elements

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

5.2.5. Tests

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 test 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: check functions.

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. Theoretical basis for Dependencies

5.3.1. Dependencies of the SFRs

[Table 30] below shows dependencies of functional components.

No	SFR	Dependencies	Reference No	SFR Type
1	FAU_ARP.1	FAU_SAA.1	3	Mandatory
2	FAU_GEN.1	FTP_STM.1	Theoretical basis(1)	Mandatory
3	FAU_SAA.1	FAU_GEN.1	2	Mandatory
4	FAU_SAR.1	FAU_GEN.1	2	Mandatory
5	FAU_SAR.3	FAU_SAR.1	4	Mandatory
6	FAU_STG.1	FAU_GEN.1	2	Mandatory
		FTP_ITC.1	Theoretical basis(2)	
7	FAU_STG.2	FAU_GEN.1	2	Conditional Mandatory
8	FAU_STG.4	FAU_STG.2	Theoretical basis(3)	Conditional Mandatory
9	FAU_STG.5	FAU_STG.2	Theoretical basis(3)	Conditional Mandatory
10	FCS_CKM.1(1)	[FCS_CKM.2 or FCS_CKM.5 or FCS_COP.1]	12, 13, 15	Mandatory
		[FCS_RBG.1 or FCS_RNG.1]	17	
		FCS_CKM.6	14	
11	FCS.CKM.1(2)	[FCS_CKM.2 or FCS_CKM.5 or FCS_COP.1]	12, 13, 16	Mandatory
		[FCS_RBG.1 or FCS_RNG.1]	17	
		FCS_CKM.6	14	
12	FCS_CKM.2	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5]	10, 11, 13	Optional
13	FCS.CKM.5	[FCS_CKM.2 or FCS_COP.1]	12, 15, 16	Conditional Mandatory
		FCS_CKM.6	14	
14	FCS.CKM.6	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5]	10, 11, 13	Mandatory
15	FCS_COP.1(1)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5]	10, 13	Mandatory
		FCS_CKM.6	14	Mandatory
16	FCS_COP.1(2)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5]	11, 13	Mandatory
		FCS_CKM.6	14	Mandatory
17	FCS_RBG.1	[FCS_RBG.2 or FCS_RBG.3]	Theoretical basis(4)	Mandatory
		FPT_FLS.1	34	
		FPT_TST.1	37	

18	FCS_RBG.4	FCS_RBG.1	17	Conditional Mandatory
		FCS_RBG.5	19	
19	FCS_RBG.5	FCS_RBG.1	17	Conditional Mandatory
		FCS_RBG.4	18	
		[FCS_RBG.2 or FCS_RBG.3 or FCS_RBG.4]	18	
20	FDP_UDE.1	FCS_COP.1	15	Mandatory
21	FDP_RIP.1	-	-	Mandatory
22	FIA_AFL.1	FIA_UAU.1	25	Mandatory
23	FIA_IMA.1	-	-	Mandatory
24	FIA_SOS.1	-	-	Mandatory
25	FIA_UAU.1	FIA_UID.1	28	Mandatory
26	FIA_UAU.4	-	-	Mandatory
27	FIA_UAU.7	FIA_UAU.1	25	Mandatory
28	FIA_UID.1	-	-	Mandatory
29	FMT_MOF.1	FMT_SMF.1	32	Mandatory
		FMT_SMR.1	33	Mandatory
30	FMT_MTD.1	FMT_SMF.1	32	Mandatory
		FMT_SMR.1	33	Mandatory
31	FMT_PWD.1	FMT_SMF.1	32	Mandatory
		FMT_SMR.1	33	Mandatory
32	FMT_SMF.1	-	-	Mandatory
33	FMT_SMR.1	FIA_UID.1	28	Mandatory
34	FPT_FLS.1	-	-	Mandatory
35	FPT_ITT.1	-	-	Mandatory
36	FPT_PST.1	-	-	Mandatory
37	FPT_TST.1	-	-	Mandatory
38	FTA_MCS.2	FIA_UID.1	28	Mandatory
39	FTA_SSL.3	FMT_SMR.1	33	Conditional Mandatory
40	FTA_TSE.1(1)	-	-	Mandatory

[Table 30] Response to security objectives and SFRs

Theoretical basis (1): FAU_GEN.1 is dependent on FPT_STM.1, which records audit events using reliable timestamps provided by TOE's operational environment, so it is satisfied by the OE. timestamps for security purposes for the operational environment.

Theoretical basis (2): FAU_STG.1 has a dependency on FTP_ITC.1 but has not been added to this protection profile because it stores audit data in local storage.

Theoretical basis (3): FAU_STG.4 and FAU_STG.5 have a dependent relationship with FAU_STG.2, which protects audit data using DBMS provided by the TOE operation environment, so it is satisfied by OE. audit data protection for security purposes for the operation environment.

Theoretical basis (4): FCS_RBG.1 has a dependent relationship with FCS_RBG.2 and FCS_RBG.3, but is satisfied

by FCS_RBG.4 and is therefore not added to this protection profile.

5.3.2. Dependency of Warranty Requirement

Since the dependent relationship of the EAL1 warranty package provided by the common evaluation criteria for the information protection system is already satisfied, the theoretical basis for this is omitted.
The added warranty requirement, ATE_FUN.1 includes ATE_COV.1 as a dependent relationship. ATE_FUN.1 was added to ensure that the developer performed the test on the test item accurately and recorded it in the test sheet, and was not added to this Security Target Statement because ATE_COV.1 indicating consistency between the test item and the TSFI was not necessarily required.

This security target specification complies with the EAL1 warranty package, but ASE_OBJ.1 includes ASE_SPD.1 that is absent from the EAL1 warranty package as a dependent relationship. However, this security target specification includes a security problem definition, and ASE_OBJ.1 performs indirect guarantees for security problem definitions, such as requiring an investigation to determine whether the security purpose of the TOE operating environment is traced to the security problem definition. Therefore, ASE_SPD.1 related to the security issue definition explanation request was not necessarily required and was not added to this security target specification.

5.4. Rationale for Security Requirements

5.4.1. Rationale for Security Feature Requirements

Rationale for the security feature requirements demonstrates the following.
Each threat and organization's security policy is handled by at least one security feature requirement.
Each security feature requirement tracks at least one threat or an organization's security policy.

Security Feature Requirement	T. Session HIJACK	T. Retry auth attempt	T Disguise	T. Reuse	T. Weak password	T Unauthorized Information Leakage.	T. Stored Data Leakage	T. Transmission Data Damage	T. Weak Cryptographic Protocol	T. TSF Damage	P. Audit	P. Safe Operation	P. Cryptostrength
FAU_ARP.1										O			
FAU_GEN.1											O		
FAU_SAA.1										O			
FAU_SAR.1											O		
FAU_SAR.3											O		
FAU_STG.1											O		
FAU_STG.2											O		
FAU_STG.4											O		

FAU_STG.5											O		
FCS_CKM.1(1)						O							O
FCS_CKM.1(2)							O	O	O				O
FCS_CKM.2						O	O	O	O				O
FCS_CKM.5						O	O	O	O				O
FCS_CKM.6						O	O	O	O				O
FCS_COP.1(1)						O							O
FCS_COP.1(2)							O	O	O				O
FCS_RBG.1						O	O	O	O				O
FCS_RBG.4						O	O	O	O				O
FCS_RBG.5						O	O	O	O				O
FDP_UDE.1						O							
FDP_RIP.1						O							
FIA_AFL.1		O	O							O			
FIA_IMA.1			O										
FIA_SOS.1				O									
FIA_UAU.1			O							O			
FIA_UAU.4			O	O						O			
FIA_UAU.7			O		O					O			
FIA_UID.1			O							O			
FMT_MOF.1										O		O	
FMT_MTD.1										O		O	
FMT_PWD.1					O					O		O	
FMT_SMF.1										O		O	
FMT_SMR.1										O		O	
FPT_FLS.1						O	O	O	O				O
FPT_ITT.1								O					
FPT_PST.1							O						
FPT_TST.1						O	O	O	O	O			O
FTA_MCS.2	O												
FTA_SSL.3	O												
FTA_TSE.1(1)	O												

[Table 31] Defining security issues and responding to security functional requirements

T.Session HIJACK FTA_MCS.2, FTA_SSL.1, FTA_SSL.3, FTA_TSE.1(1), FTA_TSE.1(2)

FTA_MCS.2 corresponds to T. session HIJACK because it restricts duplicate access to TOE with the same user account or the same authority.

FTA_SSL.1, FTA_SSL.3 correspond to T.session HIJACK because it ensures session locking or session termination for an interactive session after the inactivity period of the authorized user

FTA_TSE.1(1), FTA_TSE.1(2) ensure that authorized user access sessions are established based on IP, etc

Therefore, it responds to T.session HIJACK.

T.Retry auth attempt FIA_AFL.1

FIA_AFL.1 defines the number of failed authentication attempts by authorized managers and general users, and guarantees the ability to take responsive action when the defined number of times is reached, so it responds to T.Retry auth attempt.

T.Disguise FIA_AFL.1, FIA_IMA.1, FIA_UAU.1, FIA_UAU.4, FIA_UAU.5, FIA_UAU.7,
FIA_UID.1,
FPT_LEE.1

FIA_AFL.1 defines the number of failed authentication attempts by authorized managers and guarantees the ability to take responsive action when the defined number of times is reached, so it corresponds to T.Disguise.

FIA_IMA.1 corresponds to T.Disguise because it ensures that mutual authentication between TOE components is performed.

FIA_UAU.1, FIA_UAU.4, FIA_UAU.5, and FPT_LEE.1 correspond to T.Disguise as they ensure successful authentication of administrators, general users seeking access to TOE.

FIA_UAU.7 responds to T. Disguise because it ensures that only masked values are output or not displayed to the user during authentication, and that it does not provide feedback on the reason for failure in the event of authentication failure.

FIA_UID.1 responds to T.Disguise as it ensures successful identification of administrators, general users seeking access to TOE

T.Reuse FIA_UAU.4

FIA_UAU.4 ensures the ability to prevent reuse of authentication data and thus responds to T.Reuse.

T.Weak password FIA_SOS.1, FIA_UAU.7, FMT_PWD.1

FIA_UAU.7 ensures that only masked values are output or not displayed to the user during authentication, so it corresponds to T. Weak passwords.

FIA_SOS.1 corresponds to T.Weak password because it verifies that the password complexity rule is satisfied.

FMT_PWD.1 basically guarantees the ability to force change upon first-time access by authorized administrators to provided default passwords, so it corresponds to T.Weak passwords.

T.Unauthorized information leakage
FCS_CKM.1(1), FCS_CKM.2, FCS_CKM.5, FCS_CKM.6, FCS_COP.1(1),
FCS_RBG.1, CS_RBG.2, FCS_RBG.3, FCS_RBG.4, FCS_RBG.5,
FDP_UDE.1, FDP_RIP.1,
FPT_FLS.1, FPT_TST.1

FCS_CKM.1(1), FCS_CKM.2, FCS_CKM.5, FCS_RBG.1, FCS_RBG.2, FCS_RBG.3, FCS_RBG.4, FCS_RBG.5, FPT_FLS.1, FPT_TST.1 correspond to T.Unauthorized information leakage because it ensures that encryption

keys are generated and distributed according to secure encryption algorithms and key lengths when encrypting and decrypting user important information stored in the database.
FCS_CKM.6 responds to T.Unauthorized information leakage because it guarantees that the encryption key and related information will be destroyed according to the specified encryption key destruction method after encryption/decryption of user important information stored in the database
FCS_COP.1(1) responds to T.Unauthorized information leakage because it guarantees that cryptographic operations are performed according to the specified secure algorithm and specified encryption key length when encrypting and decrypting user important information stored in the database.
FDP_RIP.1 responds to T.Unauthorized information leakage because it ensures that all original user data will be deleted after encryption/decryption of user critical information stored in the database.
FDP_UDE.1 corresponds to T.Unauthorized information leakage because it ensures encryption/decryption performance when authorized users store critical information in the database.

T.Stored data leakage FCS_CKM.1(2), FCS_CKM.2, FCS_CKM.5, FCS_CKM.6, FCS_COP.1(2),
FCS_RBG.1, FCS_RBG.2, FCS_RBG.3, FCS_RBG.4, FCS_RBG.5
FPT_FLS.1, FPT_PST.1, FPT_TST.1
FCS_CKM.1(2), FCS_CKM.2, FCS_CKM.5, FCS_RBG.1, FCS_RBG.2, FCS_RBG.3, FCS_RBG.4, FCS_RBG.5,
FPT_FLS.1, FPT_TST.1 correspond to T.storage data leakage because it ensures that encryption keys are generated and distributed according to a secure encryption algorithm and key length when encrypting stored data.
FCS_CKM.6 responds to T.storage data leakage because it guarantees that the encryption key and its related information will be destroyed according to the specified encryption key destruction method at the end of storage data encryption.
FCS_COP.1(2) responds to T.storage data leakage because it guarantees that encryption operations are performed according to the specified secure algorithm and specified encryption key length when encrypting stored data.
FPT_PST.1 responds to T.storage data leakage because it ensures that stored TSF data is protected from leakage threats by means of encryption, access control, etc.

T.Transmission Data Damage FCS_CKM.1(2), FCS_CKM.2, FCS_CKM.5, FCS_CKM.6, FCS_COP.1(2),
FCS_RBG.1, FCS_RBG.2, FCS_RBG.3, FCS_RBG.4, FCS_RBG.5,
FPT_FLS.1, FPT_ITT.1, FPT_TST.1,
FTP_ITC.1, FTP_TRP.1
FCS_CKM.1(2), FCS_CKM.2, FCS_CKM.5, FCS_RBG.1, FCS_RBG.2, FCS_RBG.3, FCS_RBG.4, FCS_RBG.5, FPT_FLS.1, FPT_TST.1 responds to T.Transmission data leakage because it ensures that encryption keys are generated and distributed according to secure encryption algorithms and key lengths when encrypting stored data
FCS_CKM.6 responds to T.Transmission data damage because it guarantees that the encryption key and its related information will be destroyed according to the specified encryption key destruction method at the

end of encryption communication.
FCS_COP.1 responds to T.Transmission data damage because it guarantees that encryption operations are performed according to the specified secure algorithm and specified encryption key length during encryption communication.
FPT_ITT.1 responds to T.Transmission data corruption by ensuring confidentiality and integrity of data transferred between TOE components..
FTP_ITC.1 responds to T.Transmission data corruption because it ensures confidentiality and integrity of data transmitted between TOE and external IT entities.
FTP_TRP.1 responds to transmission data damage by ensuring confidentiality and integrity of data transmitted between external IT entities and TOEs in the management access process.

T.Weak cryptographic protocol FCS_CKM.1(2), FCS_CKM.2, FCS_CKM.5, FCS_CKM.6, FCS_COP.1(2), FCS_RBG.1, FCS_RBG.2, FCS_RBG.3, FCS_RBG.4, FCS_RBG.5 FPT_FLS.1, FPT_TST.1

FCS_CKM.1(2), FCS_CKM.2, FCS_CKM.5, FCS_RBG.1, FCS_RBG.2, FCS_RBG.3, FCS_RBG.4, FCS_RBG.5, FPT_FLS.1, FPT_TST.1 responds to T.Weak cryptographic protocols because it ensures that encryption keys are generated and distributed according to the encryption algorithm and key length required for standard cryptographic algorithms with a security strength of 112 bits or more when encrypting transmission data.
FCS_CKM.6 responds to T.Weak cryptographic protocols because it ensures that encryption keys and their related information are destroyed according to the specified destruction method.
FCS_COP.1(2) responds to T.Weak encryption protocols because it guarantees that encryption operations are performed according to the length of the encryption key and standard encryption algorithms with a security strength of 112 bits or more when encrypting transmission data.

T.TSF Damage FAU_ARP.1, FAU_SAA.1,
FIA_AFL.1, FIA_UAU.1, FIA_UAU.4, FIA_UAU.5, FIA_UAU.7, FIA_UID.1
FMT_MOF.1, FMT_MTD.1, FMT_PWD.1, FMT_SMF.1, FMT_SMR.1,
FPT_LEE.1, FPT_RCV.1, FPT_RCV.2, FPT_TST.1, FPT_TUD.1

FAU_ARP.1 responds to T.TSF damage because it guarantees the ability to take responsive action when detecting security violations such as TOE integrity damage.
FAU_SAA.1 responds to T.TSF damage as it ensures the ability to examine audited incidents to point out security violations such as TOE integrity damage.
FIA_AFL.1, FIA_UAU.1, FIA_UAU.4, FIA_UAU.5, FIA_UAU.7, FIA_UID.1, FPT_LEE.1 responds to T.TSF damage by allowing access to TOE only after the user's identification and authentication are successful, ensuring the blocking of bypass access by threat sources.
FMT_MOF.1, FMT_MTD.1, FMT_PWD.1, FMT_SMF.1, FMT_SMR.1 responds to T.TSF damage because it divides authorized user roles into managers and general users when accessing and setting management functions and provides security policies and security functions according to roles to prevent unauthorized access by threats.

FPT_RCV.1 responds to T.TSF Damage to ensure the ability of TOE agents or clients to recover tampered information.
FPT_RCV.2 responds to T.TSF Damage because it 'ensures that the existing version is retained automatically in case of TOE update installation fails'.
FPT_TST.1 responds to T.TSF Damage by ensuring TSF self-testing for accurate operation of TOE and the ability of authorized administrators to verify TSF data and integrity of TSF.
FPT_TUD.1 responds to T.TSF Damage because it ensures the installation and application of only validated TOE update files'

P.Audit FAU_GEN.1, FAU_SAR.1, FAU_SAR.3, FAU_STG.1, FAU_STG.2, FAU_STG.4,
FAU_STG.5,
FPT_STM.1

FAU_GEN.1 ensures that audit records are generated for auditable events such as start/end of audit functions, identification and authentication success/failure for managers, etc., thereby satisfying P.Audit.
FAU_SAR.1 satisfies P.Audit by providing authorized administrators with the ability to inquire audit records and ensuring that administrators provide audit records to suit their interpretation of information.
FAU_SAR.3 satisfies P.Audit 'because it provides an optional audit review function based on logical relationship criteria for audit data'.
FAU_STG.1 satisfies P.Audit by providing the ability to store audit data in local storage or transmitting it to an external IT entity in real time using a secure channel for TOE servers.
FAU_STG.2 satisfies P.Audit 'because it provides the function of protecting unauthorized changes and deletion of stored audit data from occurring.
FAU_STG.4 satisfies P.Audit by ensuring that appropriate response actions are performed when the audit trail of the TOE server exceeds the limit of the storage space.
FAU_STG.5 satisfies P.Audit 'because it guarantees the ability of the TOE server to take appropriate responsive action in the event of audit trail saturation'.
FPT_STM.1 satisfies P.Audit because it ensures that each component of the TOE generates an audit record using trusted time information.

P.Safe operation FMT_MOF.1, FMT_MTD.1, FMT_PWD.1, FMT_SMF.1, FMT_SMR.1
FMT_MOF.1 satisfies P.safe operation as it guarantees the ability to manage security functions only to authorized users.
FMT_MTD.1 satisfies P.safe operation as it guarantees the ability to manage TSF data only to authenticated users.
FMT_PWD.1 satisfies P.safe operation as it guarantees only authorized managers the ability to manage the combination rules and lengths of IDs and passwords, and provides functions such as password change when authorized users access for the first time.
FMT_SMF.1 satisfies P.safe operation because it 'requires TSF to specify management functions such as

security functions, security attributes, and TSF data that must be performed'.
FMT_SMR.1 satisfies P.safe operation as it guarantees specifying authorized roles related to security management.

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

FCS_CKM.1(1), FCS_CKM.1(2), FCS_CKM.2, FCS_CKM.5, FCS_CKM.6, FCS_COP.1, FCS_RBG.1, FCS_RBG.2, FCS_RBG.3, FCS_RBG.4, FCS_RBG.5, FPT_FLS.1, FPT_TST.1 satisfy P.Crypto strength because they ensure that encryption keys required for standard encryption algorithms with a security strength of 112 bits or more are generated and distributed safely during data encryption.
FCS_COP.1(1) and FCS_COP.1(2) satisfy P.Crypto strength because they ensure that encryption operations are performed according to standard encryption algorithms with a security strength of 112 bits or more and the length of the encryption key during data encryption.

6. TOE Summary Specification

This chapter outlines the security features required by the TOE.

6.1. Security Audit (FAU)

6.1.1. Audit Data Generation.

TOE generates audit data for each component (Console, EOC, KMS) and transmits it to KMS, and KMS stores audit data collected from each component in DBMS. When KMS audit data is stored, each record is stored in the local DBMS audit data table, and the audit record includes the date and time of the event, the type of the event, the identity of the subject, and the event result.

When generating audit data, TOE can selectively generate audit data for each event type (whether cipher text is included, encryption/decryption success log), and the default value generates an audit log for all audit data. For specific types of audit data, refer to the events subject to audit in Table 16.

SFR to be satisfied: FAU_GEN.1

6.1.2. Security Alarms

TOE sends an alert mail to the administrator via registered mail if a log of the administrator continuous authentication failure, integrity violation event, and failure of self-test of the KCMVP indicates a potential security violation.

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

6.1.3. Audit Review

TOE stores audit data in a database format in DBMS and only authorized administrators can view audit data through Console. When inquiring audit data, the Console can perform mutual authentication with KMS and then inquire audit data stored in DBMS through KMS.

Audit data can be viewed in detail within the log through the security audit function of the Console, and authorized administrators can selectively check the accumulated audit data for each audit data type. The following table shows how to select/order audit data by type and condition.

Audit data can protect the deletion or change of unauthorized users by using the identification and authentication functions of DBMS

Audit data type	Condition (AND)	To select or order
Administrator log	Date and time	Sort the search results that meet the selected search conditions in chronological order.
	Result Message	

[Table 32] Audit data search

SFR to be satisfied: FAU_SAR.1, FAU_SAR.3, FAU_STG.1, FAU_STG.2

6.1.4. Prevention of audit data loss

The TOE periodically monitors the audit data store to generate an audit log for exceeding the threshold when 90% of the specified audit logs are reached and sends an alert email to the authorized administrator. If the audit data store is 100% saturated, an audit log for the storage saturation is generated and an alert mail is sent to the authorized administrator via email. TOE provides a function to store the latest audit data by overwriting the oldest audit data.

SFR to be satisfied: FAU_STG.4, FAU_STG.5

6.2. Cryptographic Support

6.2.1. Cryptographic Key Generation

TOE generates random numbers from the random number generator (HASH_DRBG 256) via the Korea Cryptographic Module Validation Program (KCMVP) and optionally generates 256 bits depending on the length of the encryption key selected by the administrator. The KEK generation generates 256 bit keys through password-based encryption key guidance (PBKDF2) according to the PKCS#5 standard. TOE also generates a public key (2048 bit)/ private key pair via the Korea Cryptographic Module Validation Program (KCMVP).
For key generation, use the following by KCMVP.

Classification	Description
Cryptographic Module Name	OWLCrypto V1.0
Developed Company	OWL Systems Inc
Validation No	CM-241-2028.12
Module Type	S/W(Library)
Validation Date	Dec 22, 2023
Effective Expiration Date	Dec 22, 2028

[Table 33] KCMVP

Standard	Key Generation Algorithm	Cryptographic Key Length	Encryption Key Usage
KS X 1213-1	ARIA	256	Data Encryption Key
ISO/IEC 10118-3	SHA-256, SHA-512	N/A	

TTAK.KO-12.0331	HASH_DRBG (SHA 256)	256	Encryption and decryption of configuration Encryption and decryption of security policy file, transport data, and DB encryption key, Session key distribution, Encryption and decryption of backup file
TTAK.KO-12.0334-Part1/2 (2018)	PBKDF	256	Encryption and decryption Master key

[Table 34] TOE Cipher Key

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

6.2.2. Cryptographic Key Distribution

TOE distributes session keys for protection of transmission data between TOE components. The distributed session keys perform data protection through ARIA-256_GCM_SHA384. The distribution of session keys is based on ECDHE.

SFR to be satisfied: FCS_CKM.2

6.2.3. Cryptographic Key Inducement

TOE induces a Key Encryption Key (KEK) from password input according to the specified key derivation algorithm PBKDF conforming to TTAK.KO-12.0334-Part1/2 (2018) and the specified encryption key length of 256 bits.

6.2.4. Cryptographic Key Destruction

After using the key encryption key (KEK), KEK-inducing password, user data encryption key, TSF data encryption key, and certificate private key, TOE overwrites the encryption key memory area three times with "0" and releases the memory to safely destroy it.
SFR to be satisfied: FCS_CKM.6

6.2.5. Cryptographic Operation

When encrypting and decrypting user data stored in the DBMS that TOE wants to protect, the encryption operation is performed using ARIA-256 of the cryptographic module that has been verified. In addition, it provides user data encryption using a one-way encryption algorithm such as SHA-256, 512.
When encrypting TSF transmission data, encryption and decryption are performed using the ARIA-256 algorithm of the The validated cryptographic module(KCMVP), and when encrypting TSF storage data,

encryption and decryption are possible with ARIA-256 . In addition, integrity data is encrypted with RSA-PSS and SHA-256 and authentication data is encrypted with RSA-PSS and used.
The summary of the algorithm, encryption key length, operation mode, and operation list used in the encryption operation is shown in the following table.

Division	Standard	Cryptographic algorithm	Encryption key length	Operational mode	Computing List
Encrypt user data	KS X 1213-1	ARIA	256	CBC	DB Storage User Data Encryption Decryption
	ISO/IEC 10118-3	SHA-256, SHA-512	N/A	N/A	DB Storage User Data Encryption
Encrypt TSF data	KS X 1213-1	ARIA	256	CBC	Policy file, Transmission data, DB Encryption key Encryption Decryption, Encryption and decryption of backup file
	ISO/IEC 14888-2	RSA-PSS	2048	N/A	Mutual Authentication, Module and Configuration Integrity verification
	ISO/IEC 10118-3	SHA-256	N/A	N/A	Policy File Intergrity Verification, Administrator password encryption

[Table 35] TOE Cryptographic Operation

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

6.2.6. Random number generation

When a random number is used in an SFR that requires the use of a verification target encryption algorithm of a The validated cryptographic module(KCMVP), such as encryption key generation, a random number is generated using HASH_DRBG (SHA256) that conforms to the random number generator algorithm KS X ISO/IEC 18031 of the verification-filled cryptographic module.
Details of the The validated cryptographic module(KCMVP) that provides the random number generator used by TOE are as follows.

Classification	Description
Cryptographic Module Name	OWLCrypto V1.0
Developed Company	Owl Systems Inc
Validation No	CM-241-2028.12

Module Type	S/W(Library)
Validation Date	Dec 22, 2023
Effective Expiration Date	Dec 22, 2028

[Table 36] KCMVP

The list of noise sources included in the entropy source and the entropy provided by each noise source, as well as the seed composition and the entropy of the seed, are shown in the following table

Noise Source 1	Entropy per 1byte (bit)	Noise Source 2	Entropy per 1byte (bit)	Total Collection Lenth(byte)	Total Entropy (bit)
/dev/urandom	6.3918	time jitter (rdtsc)	5.9172	Noise source 1(32) + Noise source 2 (32) = 64	393.888

[Table 37] The list of noise sources and seed composition

In the case of /dev/urandom, the noise source output collection and configuration of the entropy source collects the pseudo-random number provided by the operating system by opening the corresponding file and reading the data. In the case of the time jitter method, after intentionally increasing the memory usage randomly using the pseudo-random number, the difference between the timestamp counter values before and after memory usage is calculated.

The noise source health test is performed on all noise sources by repeated times test (RCT) and adaptability ratio test (APT), respectively. Since the finally collected seed has sufficient entropy, the conditioning process is omitted. The entropy source output configuration is collected in 48 bytes from each entropy source and is output in binary data format.

From each entropy source data of 48 bytes that passed the noise source health test, only 32 bytes are obtained and connected to combine each entropy input. Simple connections are used as a combination method because each data provides sufficient entropy and independence is maintained regardless of the noise sources.

SFR to be satisfied: FCS_RBG.1, FCS_RBG.4, FCS_RBG.5

6.3. User data protection

6.3.1. Encrypt and decrypt in DB data

It provides a column-specific encryption/decryption method for user data stored in the DBMS that TOE wants to protect, and performs encryption/decryption in the web application server or DBMS according to the API and Plug-In method.

After the TOE encrypts and decrypts the user data, the remaining information is overwritten with "0" three times on the memory to protect the remaining information on the original data, and the memory is released and completely destroyed.

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

6.4. Identification and Authentication

6.4.1. Administrator identification and authentication

TOE provides identification and authentication functions based on the manager's ID and PW through the Console, and additionally provides authentication functions through registered certificates. Only certificate generation can be performed before manager identification and authentication. If successive authentication failures (five consecutive times) occur by the administrator while performing authentication, the TOE limits the login function of the ID for 5 minutes so that the authentication function is no longer possible.

TOE blocks information exposed on the screen by masking ('*') secret information such as passwords input during administrator authentication, and does not provide an accurate reason for authentication failure so that the password cannot be inferred through pop-up messages that occur when authentication fails.

The password required for manager authentication must be composed of a combination of at least 10 digits, 40 characters or less, numbers, uppercase letters (English), lowercase letters (English), and special characters according to the predefined combination rules to successfully authenticate. Setting a password that is identical to the user account (ID) is prohibited. The repeated use of the same character or number in succession is not allowed. Sequential input of characters or numbers based on keyboard layout is also prohibited. Reusing the most recently used password is not allowed either. In addition, in order to prevent the reuse of authentication data while the manager is authenticated, the sequence number is used to verify and prevent the reused data.

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

6.4.2. Mutual authentication between TOE components (extended)

The TOE performs mutual authentication among its components KMS, EOC, and Console using a certificate-based proprietary authentication protocol.
After basic protection of internal TSF data transmission is established through TLS, EOC_API, EOC_Plug-in, and Console request the certificate from KMS, and KMS provides its certificate to each of these components.
Each component (EOC_API, EOC_Plug-in, and Console) verifies the received KMS certificate using a locally stored Root certificate to determine its trustworthiness.
If the KMS certificate is successfully validated, each component sends its own certificate to KMS.

KMS then verifies the received certificates from the components using the same Root certificate.
Once verification is successfully completed, mutual authentication is established.

SFR to be satisfied: FIA_IMA.1(extended)

6.5. Security Management

6.5.1. Management of Security Features

The Console enables TOE to successfully authenticate based on ID and PW so that only logged in authorized administrators can perform security management functions. TOE provides security function management, TSF data management, ID and password management functions, and only authorized administrators play roles.

The administrator ID and password must be set when KMS is first operated, and the administrator password must be changed when the console is first logged in.

The security functions and administrative actions that an authorized administrator can manage are as follows:

Administrator Type	Classification	Security Function	Management Behavior Capability			
			Determine the behavior	Stop	Start	Change the behavior
Authorized Administrator	Encryption key	Creation an encryption / decryption key	○	–	○	–
	Security policy	Cipher target type	○	○	○	○
		Cipher Algorithm Type	○	○	○	○
		User data integrity check function	○	○	○	○
		Double Encryption	○	○	○	○
		Encryption Pattern	○	○	○	○
	Access Authority	User Access Authority	○	○	○	○
	Setting	User Access Authority Allow /	○	○	–	○

		Reject Policy				
		Key Policy backup	○	–	○	–
		Administrator IP Setting	–	○	○	○
		Mail Server Setting	–	○	○	○
	Audit Log	Determining Audit Target(Cipher text)	○	○	○	○
		Generating Success Log	○	○	○	○

[Table 38] The security functions and management behavior capabilities that an authorized manager can manage

In addition, the types and management capabilities of TSF data managed by authorized administrators are as follows.

Administrator	Type	Data Type	Management Behavior Capability				
			Changing Defaults	Inquiry	Changing	Creation	Deletion
Authorized Administrator	Key	User Data En/Decryption Key	–	○	–	○	○
		Master Key	–	–	○	○	–
	User	DB User	–	○	○	○	○
	Security Policy	User Data Cryptographic Policy	○	○	○	○	○
	Access Authority	Access Time	○	○	–	○	○
		Access User	○	○	–	○	○
		Access IP	–	○	○	○	○
		Access Application	–	○	○	○	○
	Configuration	Key Policy backup	–	○	–	○	–
		Admin IP	–	○	○	○	○

		Mail Server	–	○	○	○	○
	Audit Log	Admin Log	–	○	–	–	–
		Encryption Log	–	○	–	–	–
	Authentication Information	Password	–	–	○	○	–

[Table 39] The types and management capabilities of TSF data managed by authorized administrator

Guide at initial setting to set the password to a length of at least 10 digits with a combination of English/numerical/special characters, and force a reset to pop-up in case of violating this rule. Setting a password that is identical to the user account (ID) is prohibited. The repeated use of the same character or number in succession is not allowed. Sequential input of characters or numbers based on keyboard layout is also prohibited. Reusing the most recently used password is not allowed either.

[Password Combination Rule]
- Digit : more than 10 ~ less then 40 character
- numbers, uppercase letters (English), lowercase letters (English), special character include at least on each
- Prohibit setting for User Account(ID) and the same Password
- Prohibit enter the same character or number repeatedly
- Prohibit sequential input of consecutive characters or numbers on the keyboard,
- Prohibit reuse previous password.
- Number(10) : 0~9,
- Uppercase Letters(26) : A~Z,
- Lowercase Letters(26) : a~z,
- Special Character(32) : `~!@#$%^&*()-_+=[]{}₩|;:'",.<>/?

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

6.6. TSF Protection

6.6.1. Basic protection of internal transport TSF data

Cryptographic communication between TOE components uses the TLS 1.2 standard protocol to ensure safety, and the OpenSSL 3.4.1 library is applied to implementations. This process complies with the RFC 8446 standard.

AES_256_GCM_SHA384 256 bit key is used to ensure confidentiality and data integrity in cryptographic communication. The ECDHE algorithm is used in the key exchange process

The client and the server exchange temporary keys through ECDHE and generate a session key based on

the temporary keys. Thereafter, in the data transmission step, data encryption/decryption and data integrity verification are performed through the generated session key and the AES_256_GCM_SHA384 algorithm.

SFR to be satisfied: FPT_ITT.1

6.6.2. Basic Protection of Stored TSF data

Among the stored TSF data, the DB encryption key is securely encrypted (ARIA-256) and stored by the master key protected by the KMS. The master key is securely stored by encrypting with ARIA-256 using an encryption key derived from the user password, and is used to encrypt and store policy files.
The administrator password is encrypted with SHA-256 and stored in the DB, and the encryption key is overwritten three times with '0' to destroy it safely after use, so it does not exist as a plaintext.
When storing backup file, TOE settings, encryption keys, and key security parameters, encrypt with TSF data encryption keys (ARIA-256).
Whenever KMS is driven, the password required for the key derivation algorithm PBKDF is input and the master key DEK is decrypted with the derived key encryption key KEK.

SFR to be satisfied: FPT_PST.1(Extended)

6.6.3. TSF Self test

When the TOE is driven, the TOE performs its own test of the The validated cryptographic module(KCMVP) periodically (24 hours period) and performs its own test on the main TOE process. While the TOE is running, it periodically (24 hours period) performs its own test to see if the TOE process is running normally.
The TOE performs an integrity check function using the RSA-PSS algorithm on binary files such as executable files, configuration files, and library files at startup, periodically (every 24 hours), and upon administrator request.
If the TSF's own test fails, the authorized administrator will be notified by email.

Classification	Item
Process Test	pdbkms pdbwatchdog
Crypto module self test	libOWLCryptoV1.0.so
Integrity test	pdbkms pdbsysid pdbwatchdog libcrypto.so.3 libpq.so.5 libssl.so.3 owlclientsocket.so

	owlserversocket.so owltlsclientsocket.so owltlsserversocket.so pdbcert.so pdbcrypto.so accounts.dat access_user.dat access_program.dat access_ip.dat access_date.dat dbcrypt.dat policycrypt.dat usermgr.dat Master-enc.key salt.dat iv.dat pdbkms.ini pdbwatchdog.ini

[Table 40] KMS Self test target

Category	Item
Process test	pdbcallexe pdbsync
Crypto module self test	libOWLCryptoV1.0.so
Integrity test	pdbcallexe pdbsync pdbeocagent libcrypto.so.3 libpq.so.5 libssl.so.3 owlclientsocket.so owlserversocket.so owltlsclientsocket.so owltlsserversocket.so pdbcert.so pdbcrypto.so accounts.dat access_user.dat

	access_program.dat access_ip.dat access_date.dat dbcrypt.dat pdb.AccessConfig.cfg policycrypt.dat usermgr.dat Master-enc.key salt.dat iv.dat pdbcrypto.ini

[Table 41] EOC Self test target

Classification	Item
Process test	pdbconsole.exe
Crypto module self test	OWLCryptoV1.0.dll
Integrity test	pdbconsole.exe libcrypto-3-x64.dll libssl-3-x64.dll owlclientsocket.dll OWLCryptoV1.0.dll owlserversocket.dll owltlsclientsocket.dll owltlsserversocket.dll pdbcert.dll zlib1.dll pdbconsole.json

[Table 42] Console Self test target

SFR to be satisfied: FPT_TST.1, FPT_FLS.1

6.7. TOE Access

6.7.1. Admin Session Management

TOE can only perform security management functions by an authorized administrator identified and authenticated through the Console. The number of simultaneous sessions is limited to one authorized administrator based on the unique identification information and certificate of the administrator PC in which the Console is installed.

If an authorized administrator logged in through the Console has no input for 10 minutes, the session between the KMS and the Console is terminated and an administrator logout is performed automatically. TOE can only be managed by one authorized administrator, and access is not possible except for the allowed access IP. In addition, TOE terminates the previous connection when attempting simultaneous access of authorized administrators.

SFR to be satisfied: FTA_MCS.2, FTA_SSL.3, FTA_TSE.1(1)