Hancom xDB V5.0

Security Target (ST)

v1.2

HANCOM
WITH

Document Revision History

Document Versions	Notes	Date	Author(s)
v1.0	Registered a new document, Security Target: Hancom xDB V5.0	2024-05-31	KwangEun Gil
v1.1	Correction of the document in response to the request for correction of the observation report	2024-07-26	KwangEun Gil
v1.2	Modification of documents according to product function and geometry changes	2024-09-13	KwangEun Gil

contents

1. Introduction to the Security Target

This document is the Security Target (ST) of Hancom xDB V5.0 (“TOE”). It define the security functions and warranty requirements of the TOE, defines the security issues of the TOE, the security objectives, IT security requirements, and summary specifications of the TOE as the security target of Hancom xDB V5.0 (hereinafter referred to as "TOE"), the database encryption product of Hancom With Co., Ltd., and presents the theoretical basis thereof.

This Security Statement consists of the following:

- Chapter 1: Provides basic information to the TOE in the Security Target and identifies the TOE through TOE reference, TOE overview, and TOE description.
- Chapter 2: Describes the Common Criteria compliance, protection profile, and package profile of the TOE and discusses the rationale accordingly.
- Chapter 3: Describes the security objectives for the operational environment of the TOE.
- Chapter 4: Presents the definition of the extended components.
- Chapter 5: Describes the security requirements and assurance requirements for the TOE and discusses the rationale accordingly.
- Chapter 6: Describes the TOE summary specifications with regard to the security functions requirements defined in Chapter 5.

1.1. ST Reference

This Security Target is identified as follows:

Title	Security Target: Hancom xDB V5.0
ST Version	v1.2
Author(s)	Hancom With Business Support Team 2, KwangEun Gil
Creation Date	September 13, 2024
Evaluation Criteria	Common Criteria for Information Security System (Ministry of Science, ICT and Future Planning Notice (No. 2016-73))
Common Criteria Version	CC V3.1 r5
Evaluation Assurance	EAL1+ (ATE_FUN.1)

Level	
Protection Profile	National Database Encryption Protection Profile v3.0
Keywords	Encryption, Decryption, DB, Database, DBMS, Oracle

1.2. TOE Reference

TOEs that comply with this Security Objective Specification are identified as follows:

- Product name : Hancom xDB V5.0
- TOE name : Hancom xDB V5.0
- TOE version : 5.0.0.3
- TOE Components and Versions

TOE Components	version	role	Location
Hancom xDB V5.0 Policy Server	5.0.0.3	Security Policy Setting, Security Management Setting	Policy Server
Hancom xDB V5.0 API	5.0.0.3	Perform DB Encryption /Decryption	Application Server
Hancom xDB V5.0 DBAPI	5.0.0.3	Perform DB Encryption /Decryption	Database Server

[Table 1-1] TOE Components and Versions

1.3. TOE Overview

With the development of the Internet, e-commerce including e-business has become more active, and it has become more convenient to exchange and share various information, and the importance of security is also increasing. However, until now, the security of session protection and access control of the system has been the mainstay, and the stored data, which can be said to be the most important object of protection, is stored in plain text and no action is taken. Even if data is protected by using database system access control, database administrators (DBAs) have access to all data, so it is not possible to prevent information leakage by insiders. There are limitations. Therefore, there is a need for a solution that can protect the data being stored by encrypting it.

Hancom xDB V5.0 (hereinafter referred to as 'TOE') performs the function of preventing unauthorized disclosure of the information to be protected by encrypting the database (hereinafter referred to as the 'DB').

The object of encryption of TOE is the DB managed by the database management system (hereinafter referred to as 'DBMS') in the operating environment of the organization , and in this security objective specification, all data before and after it is encrypted and stored in the

DB is defined as user data. Depending on the security policy of the organization that operates the TOE, some or all of your data may be subject to encryption.
The DBMS that manages the DB in the operating environment of the organization is distinct from the DBMS that TOE directly uses to manage TSF data (security policy, audit data, etc.).
TOE consists of Hancom xDB V5.0 Policy Server 5.0.0.3 (hereinafter "Policy_ Server") installed and operated on the Policy Server, Hancom xDB V5.0 API 5.0.0.3 (hereinafter "API_ Module") installed and operated on the Application Server, and Hancom xDB V5.0 DBAPI 5.0.0.3 (hereinafter "DB_API_ Module") installed and operated on the Database Server.

1.3.1. Product Use and Main Security Characteristics

TOEs are used by authorized administrators to encrypt protected user data in order to prevent unauthorized disclosure of the information sought to be protected. Authorized administrators can decrypt and decrypt user data by setting the algorithm type and key through the encryption policy. User data subject to the encryption policy will be subject to the encryption/decryption services provided by the TOE.
The TOE includes security auditing functions that record and manage audit data for major audited incidents to enable authorized administrators to operate TOE securely within the organization's operating environment, cryptographic key management for encrypting user and TSF data, password support features such as password operations, user data protection features that encrypt user data and protect residual information, identification and authentication functions such as verification of authorized administrator identity, handling of authentication failures, and mutual authentication between TOE components; It provides security management functions for defining security functions and roles, configuration settings, TSF data protection between TOE components, TSF data protection stored in TSF regulated storage, TSF protection functions such as TSF self-examination, and TOE access functions for managing access sessions by authorized administrators.
The Data Encryption Key (DEK) used to hide and decrypt user data is encrypted and protected by the Key Encryption Key (KEK). The requirements for how to create and use DEKs and KEKs are defined as '5.1.2. Password Support (FCS)'.

The main functions of each component are as follows.

- Policy_Server

  Policy_Server will be located in an environment that uses cryptographic services. Therefore, it is independent of the environment in which the data is stored. Policy_Server receives requests for data encryption/decryption services and provides policies set by the security administrator. Policy_Server is a security management

web interface that provides the security management function of TOE to authorized administrators, and the security administrator sets the policies required by the API through Policy_Server. For this purpose, Policy_Server stores key policies, encryption keys, and audit logs in the DBMS connected to Policy_Server.

- API_Module

  The API_Module requests the policy_Server to perform the decryption by requesting the decryption policy. API_Module provides functions that can encrypt, decrypt, and digest data in order to use the encryption function on the application server. In addition, it provides libraries for various developer environments such as C and Java. Even if the API_Module is requested to provide a decryption service, the service request will be rejected if the security administrator has not set the policy or changed it for security reasons. In addition, when encryption/decryption is enabled, the audit log is sent to the P olicy_ Server.

- DBAPI_Module

  DBAPI_Module is an API module installed on the Database Server that requests the secret/decryption policy from Policy_Server to perform the encryption/decryption policy. DBAPI_Module provides functions that can encrypt, decrypt, and digest data in order to use the encryption function on the database server. It provides scripts and libraries that can be used for the target DBMS.
  Even if the DBAPI_Module requests the encryption/decryption service, if the security administrator has not set the policy or changed it for security reasons, the service request will be rejected. In addition, when encryption/decryption is enabled, the audit log is sent to the P olicy_Server.DBAPI_Module is an API method like API_ Module, which combines references to the target DBMS.

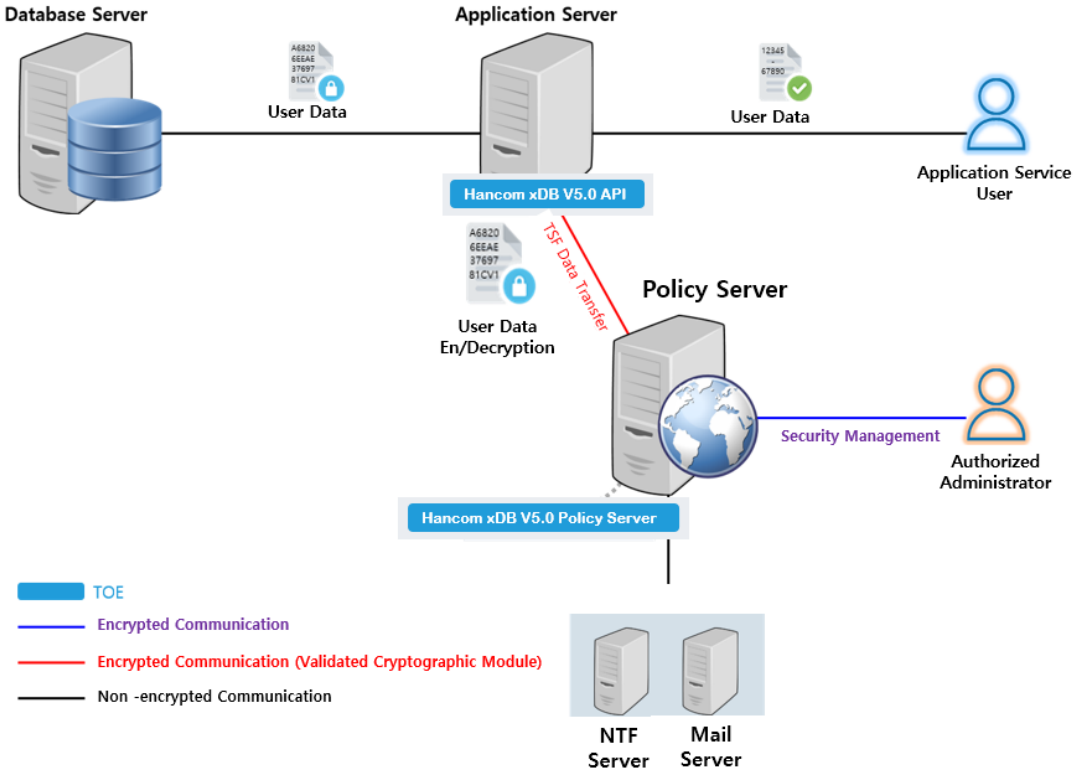
1.3.2. TOE Type

TOE is a 'DB encryption' product that performs the function of preventing unauthorized disclosure of information to be protected by encrypting a database (hereinafter referred to as 'DB'), and each component of TOE is provided in the form of software.

TOE is implemented as multiple TOE components by subdividing the roles by major functions such as the function of decryption function, audit log generation function, and the function of sending and receiving audit logs and TSF data.

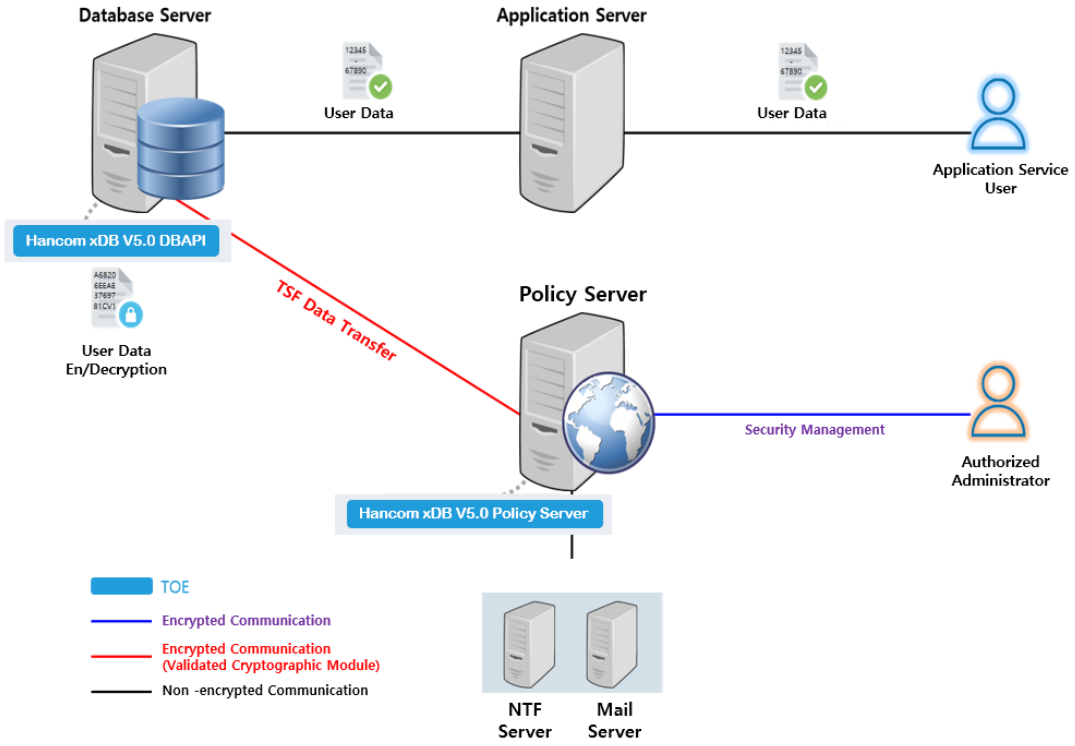
Was.

[Figure 1-1] Applications that provide application services by installing API_Module in Application Server are developed using API provided by API_Module to use TOE's encryption function. API_Module installed in Application Server performs encryption/decryption and transmits audit logs for encryption/decryption to Policy_Server. Policy_Server, which transmits security management policies for TOE and collects, records, and manages audit logs, is installed separately from API module.



[Picture 1-1] Operating Environment Configuration Chart - API Method (Management Server Separated, API_Module)

[Figure 1-2] is developed using the API provided by DBAPI_Module to use the password function of TOE on the database server when the DBAPI_Module is installed on the database server.The DB_API_ module encrypts the user data received from the Application Server before storing it in the DB according to the policy of the authorized administrator, and performs decryption of the encrypted user data when the application service user calls the user data through the Database Server.



[Picture 1-2] Operating Environment Configuration Chart - API Method (Management Server Separated, DBAPI_Module)

Authorized administrators can perform encryption and decryption of user data according to the scope of encryption required by the organization's security policy through the management server. In addition, authorized administrators can access the management server and perform security management.

The external IT entities required to operate the TOE include a mail server for administrator notification functions such as sending audit data loss prediction emails and a DBMS server for storing audit data. Except for TOE, external IT entities and business systems correspond to the operating environment of TOE.

1.3.3. Non-TOE Hardware/Software Identification

Additional hardware and software are required to operate the TOE, but they are not included in the assessment.

The hardware and software requirements to operate the TOE are as follows:

(1) Minimum System Requirements to Operate TOE

TOE	OS	Division	Minimum Requirements

Policy_Server	Linux	OS	Redhat 8.9 kernel 4.18.0 (64bit)
		CPU	Xeon E3 - 1220 3.1 G G66
		RAM	16 GB or more
		HDD	At least 100 GB of space required for TOE installation and operation
		NIC	10/100/1000 Ethernet Card 1 Port more
API_Module	Linux	OS	Redhat 8.9 kernel 4.18.0 (64bit)
		CPU	Xeon E3 - 1220 3.1 G G66
		RAM	16 GB or more
		HDD	At least 1 GB of space required for TOE installation
		NIC	10/100/1000 Ethernet Card 1 Port more
DBAPI_Module	Linux	OS	Redhat 8.9 kernel 4.18.0 (64bit)
		CPU	Xeon E3 - 1220 3.1 G G66
		RAM	16 GB or more
		HDD	At least 1 GB of space required for TOE installation
		NIC	10/100/1000 Ethernet Card 1 Port more

[Table 1-2] TOE Operating Environment Hardware Requirements

(2) Minimum Administrator System Requirements for Security Management

category	Minimum Requirements
Web Browser	Chrome 124.0

[Table 1-3] Hardware and Software Requirements of Administrator System

(3) Non-TOE software required for TOE operation

TOE	S/W	Purpose
Policy_Server	Java(JRE) 1.8.0_412	Java application-based server operation and operation, security management functions, and web server operation

	Apache Tomcat/7.0.109	Web browser on the administrator system – performs encrypted communication between servers ServerWeb server for providing security management screen
	Oracle 19c	DBMS for TOE Management
API_Module	Java(JRE) 1.8.0_412	Java Application-based TOE DB API operation and operation
DBAPI_Module	Java(JRE) 1.8.0_412	Java Application-based TOE DB API operation and operation
	Oracle 19c	DBMS for TOE installation

[Table 1-4] Identification and description of non-TOE software required for TOE operation

(4) External IT entities used other than those subject to evaluation

category	TOE Support Function
Mail Server (SMTP Server)	A server for sending mail to authorized administrators when a potential security breach is detected.
DBMS	DBMS for storing audit logs of TOE

[Table 1-5] External IT Entities

1.4. TOE Description

This chapter describes the scope and boundaries of the TOE.

1.4.1. Physical Scope of the TOE

The physical scope of TOE consists of the Policy_Server that performs security management such as policy settings, the API_Module installed in the Application Server, and the DBAPI_Module package installed in the database, as well as the manual required for installation and operation. Details are as shown in [Table 1-6].

(1) TOE Details

Division	identifier		Format	Distribution
TOE	HanacomxDB V5.0 (version : 5.0.0.3)		S/W	-
TOE Component	Hancom xDB V5.0 Policy Server 5.0.0.3(z_package. Hancom_xDB_V5.0_Policy_Server.5.0.0.3.tar.gz)			CD
	Hancom xDB V5.0 API 5.0.0.3(z_package. Hancom_xDB_V5.0_API.5.0.0.3. Linux.x86_64.64bit.tar.gz)			
	Hancom xDB V5.0 DBAPI 5.0.0.3(z_package. Hancom_xDB_V5.0_DBAPI.oracle.5.0.0.3. Linux.x86_64.64bit.tar.gz)			
Manual	Preparatory Procedure	Hancom xDB v5.0 Preparative Procedure (PRE) v1.2(Hancom xDB_V5.0_ Preparative Procedure(PRE)_v1.2.pdf)	electron Document (PDF)	
	Operating Instructions	Hancom xDB V5.0 Operation Guide (OPE) v1.2(Hancom xDB V5.0_ Operation Guide(OPE)_v1.2.pdf)		

[Table 1-6] Physical Scope of the TOE

The details of the verification cryptographic modules included in the TOE are as follows:

category	content
Cryptographic Module Name	XecureCrypto v2.1.0.0
Verification number	CM-247-2029.5
Verification Grade	VSL1

Developer	HANCOM WITH
Verification date	May 30, 2024

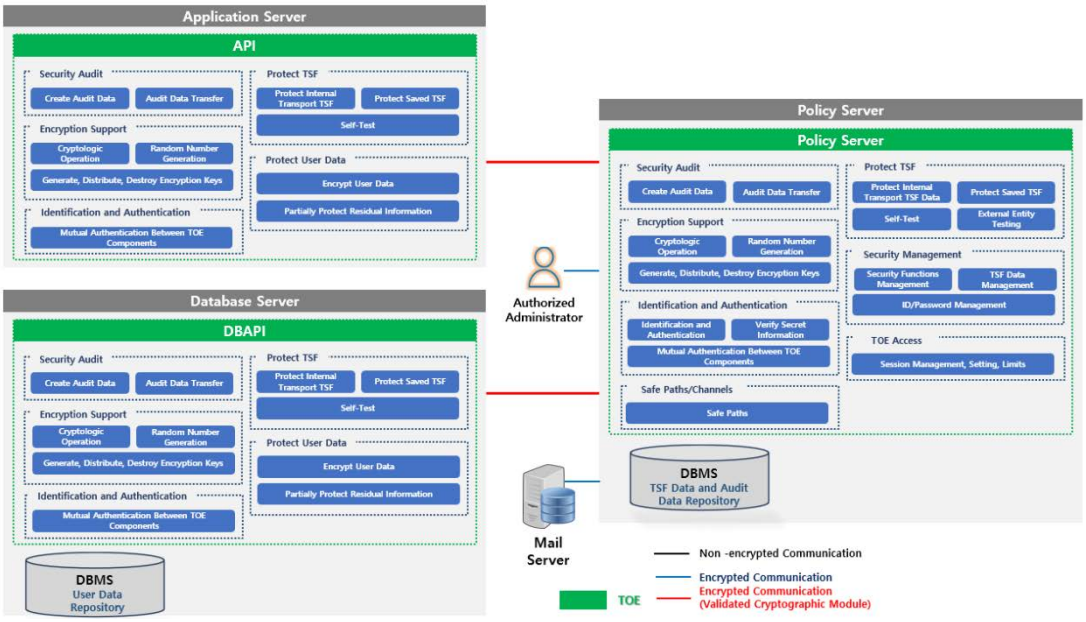
[Table 1-8] General Verification Cryptographic Module

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

Item		description
Policy_Server	Log4j 2.14.1	Web Application Server service log records

[Table 1-9] 3rd party software required for TOE operation

1.4.2. Logical Scope of the TOE



[Figure 1-3] Logical Scope of TOE

- **Security Audit**

  The TOE shall create and record audit records of the audited incidents, such as the operation of the security functions provided by the TOE and the history of security management. The information recorded when creating an audit record is the time of occurrence of the event to be audited, items, subject information such as ID or access IP, and processing results. Policy_Server, the policy server and security management server of TOE, generates audit data for audited incidents arising from the performance of administrative functions (including security management functions) and stores them

in the DBMS. In addition, the user data performed by the API module is stored in the cancer·It generates audit data for audited incidents that occur including decryption, and transmits audit data to the Policy Server. Policy_Server stores the audit data sent from the API module in the DBMS.

When the TOE detects a potential security breach, it takes action in response to the security breach. In the event of any potential breach audit, the authorized manager will be notified by email.

The TOE provides a review and selective review function for the audit data generated by the TOE and stored in the DBMS. Selectively review the stored API audit log according to the query date, encryption policy ID, encryption account, operation type, and result, and selectively review the administrator audit log according to the job type, job behavior, worker, result, view period, and CLIENT IP. Authorized administrators can perform reviews and optional reviews of each audit log.

It stores all generated audit data in the DBMS to safely manage it, prevents unauthorized deletion of audit data, and protects the audit repository by ignoring audited incidents when the audit repository is saturated.

- **Cryptographic support**

  TOE provides cryptographic key generation, destruction function, and cryptography function to protect data transmission between TOE components and to hide and decrypt user data. In addition, it provides a random number generation function for secure encryption key generation. TOE generates encryption keys for user data encryption and TSF data encryption using the cryptographic algorithm subject to verification of the verified cryptographic module 'XecureCrypto v2.1.0.0', which has been confirmed to be safe and suitable for implementation through the Cryptographic Module Verification System (KCMVP). When the encryption key is no longer needed, the encryption key is deleted and destroyed by overwriting it with zero by the method of key zeroization.

  For the encryption of user data, TOE performs cryptographic operations using the cryptographic algorithm subject to verification of the verified cryptographic module 'XecureCrypto v2.1.0.0', whose safety and suitability for implementation have been confirmed through the Cryptographic Module Verification System (KCMVP), and the cryptographic module operates in the verifiable operation mode when performing cryptographic operations. When performing encryption using a block cipher algorithm, ECB mode is not used, regardless of the size of the plaintext , and the use of IV in CBC mode is based on KS X 1213 and TTAS. Apply the method suggested in KO-12.004/R1.

  For TSF data encryption, TOE performs cryptographic operations using the

cryptographic algorithm subject to verification of the cryptographic module 'XecureCrypto v2.1.0.0', which has been confirmed for safety and suitability through the Cryptographic Module Verification System (KCMVP), and the cryptographic module operates in the verifiable operation mode when performing the cryptographic operation.

When using random numbers in SFR, which requires the use of a cryptographic algorithm to be verified by a cryptographic module to be verified, such as generating a key encryption key such as a key (DEK) for user data encryption, TOE uses a random number generator of the verifiable cryptographic module 'XecureCrypto v2.1.0.0' whose safety and suitability have been confirmed through the Cryptographic Module Verification System (KCMVP).

- Generate a Cryptographic Key
  - HASH_DRBG(SHA256, 256bit) : Encryption key generation for the encryption and decryption of TSF data, encryption and decryption of user data, and encryption and decryption of encryption keys (KEK, DEK)

- Cryptographic operations
  - Symmetrical key arm·Decryption (ARIA-CBC, 128bit): TSF data, encryption key (KEK, DEK) cancer, decryption,
  - Symmetric Key Encryption and Decryption (ARIA-CBC, 128/192/256 bit): Performs data encryption and decryption between TOE components in self-implemented communications
  - Symmetric Key Encryption and Decryption (SEED-CBC, 128 bit): Performs data encryption and decryption between TOE components in self-implemented communications
  - One-way encryption (SHA256): TSF data encryption, integrity verification
  - One-way encryption (SHA224, 256, 384, 512): Encrypts user data

- Cryptographic Key Destruction
  - The encryption key that is loaded into memory during key generation, distribution, update the temporarily stored encryption key information to 0x00 and destroy the encryption key information.

- **Protecting user data**

  TOE provides a function that can be decrypted and decrypted user data by column.

In addition, in order to prevent the same ciphertext from being generated for the same plaintext data when encrypting user data, a random initial vector (Initial Vector, IV) is used for encryption.

When allocating and retrieving resources to user data, in order to prevent all previous information contents of resources from being available, data in memory is zeroed and deleted after user data is decrypted.

- **Identification & Authentication**

  In order to allow access to the security management features provided by the TOE, the authorized administrator must be successfully identified and authenticated before all actions related to the security features are permitted.

  The identity of the authorized administrator is verified based on ID and password. The security administrator's password must be uppercase, lowercase, number, and special characters (!,@, #,%, ^,*, (,), -, =, _, +, [,], {, }; ,., /, >, ?) It should be a combination of all 4 characters and should be between 9 and 20 characters.

  When identifying and authenticating an authorized administrator's administrative access, if authentication fails 5 times in a row, the management access attempt of the account will be blocked for 5 minutes, and the audit record of the authentication failure will be stored.

  In order to protect authentication feedback, TOE masks the password entered when logging in, adding administrators, and modifying information in the TOE security management so that it cannot be seen on the screen. In addition, in the event of a failure of identification and authentication, it does not provide feedback on the reason for the failure.

  TOE ensures the uniqueness of the session ID by using a random number generated by the verifiable cryptographic module to prevent the reuse of authentication data.

  Mutual authentication between the API module and Policy_Server is performed through a self-implemented authentication protocol. The mutual authentication method issues the private key and public key of the API module and Policy_Server, respectively, and then generates a signature (api-signMessage) with a specific message (api-originMessage) and an API private key (api-privateKey) in the API module.

  And in Policy_Server, a signature (api-signMessage) is authenticated with a specific message (api-originMessage) and the API public key (api-publicKey). In the same way, a signature (pol-signMessage) is generated with a specific message (pol-originMessage) and the Policy_Server private key (pol-privateKey) in Policy_Server. And in the API module, mutual authentication is performed by authenticating a signature (pol-

signMessage) with a specific message (pol-originMessage) and the Policy_Server public key (pol-publicKey).

- **Security Management**

  The TOE provides a security management function that allows authorized administrators to set and manage security policies and important data. In addition, after installing the Policy_Server, it provides a function to forcibly change the administrator password when accessing the administrator for the first time. In order to use the security management function, the administrator must go through the identification and authentication process to use the security management function.

- **TSF Protection**

  TOE uses the cryptographic algorithm subject to verification of the cryptographic module "XecureCrypto v.2.1.0.0", which has been verified for its safety and suitability for implementation through the Cryptographic Module Verification System (KCMVP), to protect the transmitted TSF data such as audit data and critical security parameters from exposure and alteration when TSF data is transmitted between separate parts of the TOE.

  TOE protects the passwords, encryption keys, core security parameters, TOE setting values (security policy, configuration parameters), and audit data of authorized administrators and DB encrypted users stored in the TSF data store from unauthorized disclosure or alteration. In particular, TSF data such as passwords of authorized administrators and DB encryption users, Data Encryption Key (DEK), core security parameters, TOE setting values, and DBMS connection information are encrypted and stored with the verifiable cryptographic algorithm of the cryptographic module "XecureCrypto v.2.1.0.0".

  The DEK (Data Encryption Key) is encrypted and stored with the verifiable encryption algorithm provided by the verifiable cryptographic module using the key encryption key (KEK).

  The encryption key and core security parameters loaded into the memory are destroyed from memory after the decryption operation is completed.

  TOE conducts self-testing at start-up and periodically during regular operation in order to verify the correct operation of the Policy_Server.

  The core process of performing TSF is subject to self-testing, and cryptographic modules that must be verified can also receive the results of the self-test and point out potential violations through self-testing.

TOE provides TSF data such as major executable files and configuration files, as well as the function of verifying the integrity of TSF. Integrity checks shall be performed at start-up, periodically during regular operation, and at any time desired by the authorized administrator.
In addition, if integrity is compromised as a result of the integrity check, the authorized administrator will be notified by e-mail.

- **Access to TOE**
  The TOE limits the maximum number of concurrent connection sessions to 1, preventing concurrent access to the same account. The TOE also blocks simultaneous access to the same authority. In the event of simultaneous attempts to access the same account or the same authority, block the blocking of new access and maintain the existing access.
  The TOE terminates the session if there is no idle time (5 minutes) after the authorized administrator logs in.
  TOE controls access to the TOE so that only registered IPs (no more than 2 default values) can access the security management interface. After installing TOE, you can set the access allowed IP at the first login, and after installation, you can add, change, or delete the access allowed IP through the security management login allowed IP list setting. When setting up an IP, you can't specify an IP address range, and you have to add one IP address individually. In this case, it is not allowed to set 0.0.0.0, 192.168.10.*, any, etc., which means the entire network.

- **Trusted Paths/Channels**
  The TOE provides a secure channel to protect data from unauthorized alteration or disclosure when working with a mail server to send mail to an authorized administrator in the event of a potential breach.

1.5. Document Conventions

This Security Target uses English to convey some abbreviations and exact meanings. The notation, form, and rules of writing used follow the Common Criteria.

The Common Criteria specify the operations that can be performed on the security functional requirements. In this Security Target, iteration, assignment, selection, and refinement operations are used.

- **Iteration**

It is used to repeat one component several times by applying various operations. The result of the iteration operation is indicated by the iteration number in parentheses after the component identifier, i.e., (iteration number).

- **Assignment**

Used to assign a specific value to an unspecified parameter (password length, for example). The result of the assignment operation is indicated in square brackets, namely [assignment value].

- **Selection**

When describing the requirements, it is used to select one or more of the options provided in the computer security system's Common Criteria. The result of the selection operation is indicated in *underlined italics*.

- **Refinement**

It is used to limit the requirements further by adding details to the requirements. The result of the refinement operation is shown in **bold face**.

1.6. Definitions of Terms

The technical terms used in this Security Target are defined below; terms that are the same as those used in the Common Criteria are in accordance with the Common Criteria.

- **Private Key**

  Used with an asymmetric cryptographic algorithm, a cryptographic key uniquely combined with a single entity (subject using the private key); should not be disclosed

- **Object**

  Passive entity in the TOE, subject to operation of the subject; contains or receives information

- **Approved Mode of Operation**

  Operation mode of cryptographic module using the verification target cryptographic algorithm

- **Approved Cryptographic Algorithm**

  Cryptographic algorithm selected by the cryptographic module verification authority for block cipher, hash function, message authentication code, random number generator, key settings, public key cryptography, and digital signature cryptographic algorithm considering safety, reliability, and interoperability

- **Attack Potential**

  The degree of effort required to attack the TOE, identified in terms of attacker expertise, resources, motivation, etc.

- **Public Key**

  Used in conjunction with an asymmetric cryptographic algorithm, a cryptographic key uniquely combined with a single entity (subject using the public key). Can be disclosed

- **Public Key (Asymmetric) Cryptographic Algorithm**

  Cryptographic algorithms using the public key and private key pairs

- **Management Access**

  The administrator attempts to connect using HTTPS, SSH, TLS, IPSec, etc. for TOE management purposes

- **Random Bit Generator (RBG)**

  Device or algorithm that outputs a statistically independent, unbiased binary string. Random number generators used for cryptographic applications typically generate bit sequences of zeros and ones, can be combined into random blocks. Random number generators are classified into deterministic and nondeterministic methods. The deterministic random number generator consists of an algorithm that generates a string of bits from an initial value called seed key, whereas the nondeterministic random number generator produces an output that depends on an unpredictable physical source.

- **Symmetric Cryptographic Technique**

  Encryption technique using the same secret key in encryption and decryption mode; also known as secret key cryptographic technique

- **Database (DB)**

  A collection of data organized according to a certain structure to receive, store, and supply data in response to the needs of multiple users so as to support multiple application tasks at the same time. The database related to encryption by column as required in this protection profile means a relational database.

- **Data Encryption Key (DEK)**

  Key for encrypting and decrypting data

- **Iteration**

  Using the same component to express two or more different requirements

- **SFP, Security Function Policy**

  A set of rules describing the specific security actions performed by the TSF (TOE security functionality) and which can be expressed in terms of SFR (Security Functional Requirements)

- **Security Target (ST)**

  Implementation-dependent security requirement specification suitable for a specific TOE

- **Security Attribute**

  These values are used to enforce the SFR such as characteristics of the subject, user (including external IT products), objects, information, sessions, and/or resources used to define the SFR.

- **Security Token**

  Hardware device implementing key generation, digital signature generation, etc. inside the device to store secret information securely

- **Protection Profile (PP)**

  Implementation-independent security requirements specification for the TOE type

- **Decryption**

  Restoring the ciphertext to the original plaintext using a decryption key

- **Secret Key**

  Used in conjunction with a secret-key cryptographic algorithm, a cryptographic key uniquely combined with one or more entities; should not be made public

- **User**

  See "External Entities."

- **User Data**

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

- **Selection**

  Specifying one or more items from the list described in the component

- **Identity**

  Unique expression that identifies the authorized user. It may be the user's real name, abbreviation, or pseudonym.

- **Encryption**

  Converting plaintext into ciphertext using an encryption key

- **Element**

  Minimum unit of security requirements that cannot be split

- **Role**

  Set of predefined rules establishing the allowed interactions between the user and the TOE

- **Operation (on a component of the CC)**

  Modifying or repeating components. The operations allowed on a component are assignment, iteration, refinement, and selection.

- **Operation (on a subject)**

  Specific actions performed by a subject on an object

- **External Entity**

  Entity (human or IT) interacting with (or can interact with) the TOE from outside the TOE

- **Threat Agent**

  Unauthorized external entities that threaten illegal access, alteration, or deletion of assets

- **Authorized Administrator**

  Authorized user who operates and manages the TOE safely

- **Authorized User**

  Users who can execute functions in accordance with the security functional requirements (SFR)

- **Authentication Data**

  Information used to prove the user's identity

- **Self-Test**

  Pre-operational and conditional tests performed by cryptographic modules

- **Assets**

  Entity that gives value to the owner of the TOE

- **Refinement**

  Specification by adding details to a component

- **Organizational Security Policies**

  A set of security rules, procedures, practices, and guidelines that are presently and/or likely to be granted to the operating environment by real or virtual organizations.

- **Dependency**

  As a relationship between components, if the requirement based on the dependent component is included in the protection profile, security target, or package, the requirements based on the dependent component (that component) are also in the protection profile, security target, or package.

- **Subject**

  Active entity in the TOE, performs operations on objects

- **Augmentation**

  Adding one or more requirements to a package

- **Column**

  A set of data values with a specific data type corresponding to one value of each row in a relational database table

- **Component**

  The smallest selection unit that can be used to form the basis of a requirement as a set of elements

- **Class**

  Collection of Common Criteria family with the same security objectives

- **KEK, Key Encryption Key**

  Key that encrypts and decrypts another encryption key

- **TOE, Target of Evaluation**

  Software, firmware, and/or hardware set with possible documentation

- **EAL, Evaluation Assurance Level**

  Assurance package consisting of three parts of assurance requirements with a predefined assurance level in the Common Criteria

- **Family**

  A collection of components with similar purpose but differ in emphasis or rigor

- **Assignment**

  Specific specification of the parameters identified within the component or requirement (of the Common Criteria)

- **CSP, Critical Security Parameters**

  Security-related information (e.g., authentication data such as secret keys, private keys, passwords, or personal identification numbers) that can be compromised if exposed or modified

- **Application Server**

  Application Server as defined in this Protection Profile refers to a server on which an application developed to provide a specific application service in an organization that operates the TOE is installed and operated. The application reads user data from the DB existing in the database server at the request of an application service user or transmits the user data to be stored in the DB to the database server.

- **Database Server**

  Database server as defined in this Protection Profile refers to a server on which a DBMS that manages a protected DB is constructed in the organization that operates the TOE.

- **DBMS (Database Management System)**

  A software system configured to compose and apply a database. The DBMS related to column-level encryption required by this Protection Profile refers to a database management system based on a relational database model.

- **SSL (Secure Sockets Layer)**

  Security protocol proposed by Netscape to provide security, such as confidentiality and integrity in the computer network

- **TLS (Transport Layer Security)**

  A protocol for encrypted authentication communication between SSL-based servers and clients as described in RFC 2246

- **TSF, TOE Security Functionality**

  Set of all hardware, software, and firmware of the TOE, contributes to the correct performance of the SFR

- **TSF Data**

  Data generated for the TOE by the TOE, may affect the operation of the TOE

- **Tablespace Size**

Total TOE audit record storage capacity (DBMS)

2. Declaration of Compliance

The Declaration of Compliance describes the Common Criteria, Protection Profiles, and Packages that this Security Objectives Specification complies with, as well as the Safeguards Profile and how the Security Objectives Specification complies with the Safeguarding Profile.

2.1. Common Criteria Conformance

This Security Objective Specification complies with the Common Evaluation Standards for Information Security Systems (Ministry of Science, ICT and Future Planning Notice No. 2013-51) V3.1 Revision 5th Edition Part 2 and Part 3 .

Common Criteria		Common Evaluation Criteria for Information Security System Version 3.1 Revision 5th Edition - Common Evaluation Criteria for Information Security System Part 1 : Introduction and General Model, Version 3.1r5 (CCMB-2017-04-001, 2017. 4) - Common Evaluation Criteria for Information Security System Part 2: Security Function Component, Version 3.1r5 (CCMB-2017-04-002, 2017. 4) - Common Evaluation Criteria for Information Security Systems Part 3: Warranty Components, Version 3.1r5 (CCMB-2017-04-003, 2017. 4)
Protection Profiles		National Database Encryption Protection Profile v3.0
Compliance Form	**Part 2 Security functional Requirements**	Extension : FCS_RBG.1, FIA_IMA.1, FDP_UDE.1, FMT_PWD.1, FPT_PST.1
	Part 3 Security assurance Requirements	In compliance
	package	Add: Add EAL1 (ATE_FUN.1)

2.2. Protection Profile Compliance

This security objective specification complies with the same security objectives and security requirements for the operating environment by strictly complying with the 'National Database Encryption Protection Profile V3.0 (KECS-PP-1232-2023)'.

2.3. Package Compliance

The package of warranty requirements that this Security Objective Specification complies with is EAL1, which further defines some warranty requirements.

- Assurance package: Added EAL1 (ATE_FUN.1)

2.4. Rationale for the Declaration of Compliance

Since this Security Objective Specification accommodates the TOE type, security objectives, and security requirements of the Protected Profile, the declaration of compliance of the Database Encryption Protection Profile V3.0 is " Strict Protection Profile Compliance".

Rationale for the security purpose of the added operating environment

Item	Security Objectives	Rationale
Security Objectives for the Operational Environment	OE.Time stamp	Adds security objectives to the operating environment by accurately recording security-related incidents using reliable timestamps provided by the TOE operating environment.
	OE. DBMS	The audit data repository is protected through the DBMS provided by the TOE operating environment, adding a security purpose for the operating environment.
	OE. Management Access	By performing administrative access through the web browser provided by the TOE operating environment, the security purpose of the operating environment is added.

3. Security Objectives

This Security Objectives Specification defines security objectives by categorizing them into TOE security purposes and security objectives for the operating environment. The TOE security purposes are the security purposes that are directly addressed by the TOE, and the security purposes for the operating environment are the security purposes that are addressed by the IT environment or non-technical or procedural means.

3.1 Security Objectives for the Operational Environment

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

OE. Physical Security

Places where TOE is installed and operated shall be equipped with access control and protection facilities so that only authorized administrators can access them.

OE. Trusted Administrator

The authorized administrator of the TOE shall not be malicious, has been properly trained in the functions of TOE administration, and shall perform his duties exactly and in accordance with the administrator's instructions.

OE. Safe Development

Developers who use the TOE to integrate encryption into their application or DBMS must comply with the requirements of the documentation accompanying the TOE to ensure that the security features of the TOE are applied securely.

OE. Log Backup

In preparation for the loss of audit records, the authorized administrator of the TOE shall periodically check the free space in the audit data repository and perform a backup of the audit records (external log server, separate storage device, etc.) to prevent the audit records from being exhausted.

OE. Operating System Enhancement

The authorized administrator of the TOE shall ensure the reliability and safety of the operating system by mitigating the latest vulnerabilities in the operating system in which the TOE is installed and operated.

OE. Time stamp

The TOE must accurately record security-related incidents using reliable timestamps provided by the TOE operating environment.

OE. DBMS

In order to protect the storage where TSF data is stored, the DBMS should be installed and operated in such a way that all connections other than those via TOE are blocked.

OE. Administrative Access

The TOE requires that all information transmitted when an authorized administrator accesses the management server through a web browser must be protected through a secure channel/channel.

4. Extended Components Definition

In addition to the components of Part 2 of the Common Evaluation Criteria, this Security Objective Specification additionally defines and uses the following components. The extension components in this Security Objective Specification are as follows:

- Cryptographic Support (FCS)
  - FCS_RBG.1 Generating Random Numbers

- Identification & Authentication (FIA)
  - FIA_IMA.1 Mutual Authentication Between TOE Components

- User data protection (FDP)
  - FDP_UDE.1 Encryption of user data

- Security Management (FMT)
  - FMT_PWD.1 Identity and Password Management

- Protection of the TSF (FPT)
  - FPT_PST.1 Basic protection of stored TSF data

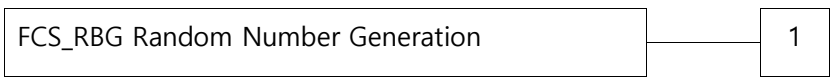
4.1. Cryptographic Support (FCS)

4.1.1. Random Number Generation

Family Overview

The Random Bit Generation (FCS_RBG) family is defined as requiring the ability to generate the random values required for TOE cryptographic operations.

Component hierarchies and explanations



FCS_RBG.1 Random number generation requires TSF to generate the random value required for TOE cryptographic computation.

Management: FCS_RBG.1
No expected management requirements

Audits: FCS_RBG.1
There are no auditable events foreseen.

FCS_RBG.1 Random number generation
Hierarchical to: No other components
Dependencies: No dependencies

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

4.2. Identification & Authentication (FIA)

4.2.1. TOE internal mutual authentication

Family Overview
The TOE internal mutual authentication (FIA_IMA) family requires that the TSF provides mutual authentication function between TOE components in the process of user identification and authentication.

Hierarchy and Description of the Component(s)

Mutual Authentication Between FIA_IMA TOE Components	1

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

Management : FIA_IMA.1
No expected management requirements 0

Audit : FIA_IMA.1

FAU_GEN If a security audit data generation family is included in the Protection Profile/Security Target, it is recommended that the following behaviors be audited:

a) Minimum: Success and failure of mutual authentication

FIA_IMA.1 Mutual Authentication Between TOE Components

Hierarchy: No other components

Dependencies: No dependencies

FIA_IMA.1.1 The TSF shall mutually authenticate between [Assignment: Separate Parts of the TOE] via [Assignment: Authentication Protocol] that conforms to [Assignment: List of Standards].

4.3. User data protection (FDP)

4.3.1. Encrypt user data

Family Overview

This family provides the requirements for ensuring the confidentiality of user data.

Component hierarchies and explanations

FDP_UDE User data encryption	1

FDP_UDE.1 User Data Encryption requires that user data be kept confidential.

Management : FIA_UDE.1

The following management functions may be considered in FMT:

a) Rule management of user data encryption and decryption

Audit : FDP_UDE.1

FAU_GEN If a security audit data generation family is included in the Protection Profile/Security Target, it is recommended that the following behaviors be audited:

a) Minimum: Success or failure of user data encryption and decryption

FDP_UDE.1 Encryption of user data

Hierarchy: No other components

Dependencies: FCS_COP.1 Cryptographic operation

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

4.4. Security Management (FMT)

4.4.1. ID & Password

Family Overview

The ID and password (FMT_PWD) family defines an authorized user to control the management of identities and passwords used in the TOE and to require the ability to set or change identities and/or passwords.

Component hierarchies and explanations

FMT_PWD ID and Password	1

FMT_PWD.1 Identity and password management requires the TSF to provide identity and password management capabilities.

Management : FMT_PWD.1

The following management functions may be considered in FMT:

a) Management of ID and password setting rules

Audit : FMT_PWD.1

FAU_GEN If a security audit data generation family is included in the Protection Profile/Security Target, it is recommended that the following behaviors be audited:

a) Minimum: Any changes to the password

FMT_PWD.1 Identity and Password Management

Hierarchy: None

Dependencies: FMT_SMF.1 Administrative Function Specification

FMT_SMR.1 Security roles

FMT_PWD.1.1 TSF *should limit the* ability to manage passwords in the Assignment *Function List to Assignments* Authorized Roles as follows:

1. [assign: *password combination rule and/or length*]
2. [Assignment: *Manage other special characters to be excluded from passwords*]

FMT_PWD.1.2 TSF *should limit the* ability to manage the IDs of the Assignment *: Feature List to the following Assignments:* Authorized Roles:

1. [assign:*ID combination rule and/or length*]
2. [Assignment: *Manage other special characters to be excluded from the ID*]

FMT_PWD.1.3 The TSF *shall provide the ability to [optional:* set an ID and password during the installation process, set a password during the installation process, change the ID and password on first connection by an authorized administrator, or change the password on first access by an authorized administrator].

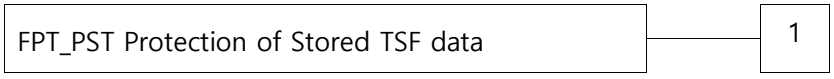
4.5. Protection of the TSF (FPT)

4.5.1. Basic Protection of Stored TSF Data

Family Overview

The Protection of Stored TSF data (FPT_PST) family defines rules to protect TSF data stored in TSF-controlled storage from unauthorized alteration or exposure.

Component hierarchies and explanations



FPT_PST.1 The basic protection of TSF data at rest requires that TSF data stored within a repository controlled by TSF be protected.

Management : FPT_PST.1

There are no management activities foreseen.

Audit : FPT_PST.1

There are no auditable events foreseen.

FPT_PST.1 Basic protection of stored TSF data

Hierarchy: No other components

Dependencies: No dependencies

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

5. Security Requirements

This chapter describes the security feature requirements and warranty requirements that the TOE must satisfy.

5.1. Security Functional Requirements

The security functional requirements defined in this Security Objective Specification are selected and used from the Common Evaluation Criteria Part 2 and Chapter 4 Extension Component Definitions.

Security Feature Classes	Security Components	
Security Audit (FAU)	FAU_ARP.1	Security Alarms
	FAU_GEN.1	Audit Data Generation
	FAU_SAA.1	Analysis of potential violations
	FAU_SAR.1	Audit Review
	FAU_SAR.3(1)	Optional Audit Review (API Audit Log)
	FAU_SAR.3(2)	Selectable Audit Review (Administrator Audit Log)
	FAU_STG.3	Responding Actions in Prediction of Audit Data Loss
	FAU_STG.4	Audit Data Loss Prevention
Password support (FCS)	FCS_CKM.1(1)	Generate encryption keys (encrypt user data)
	FCS_CKM.1(2)	Encryption Key Generation (TSF Data/Mutual Authentication Encryption)
	FCS_CKM.4	Breaking the encryption key
	FCS_COP.1(1)	Cryptography (encryption of user data)
	FCS_COP.1(2)	Cryptography (TSF Data Encryption/Mutual Authentication)
	FCS_RBG.1(Extension)	Random Number Generation
User Data Protection (FDP)	FDP_UDE.1 (Extension)	Encrypt user data
	FDP_RIP.1	Partial residual protection
Identification & Authentication (FIA)	FIA_AFL.1	Authentication Failure Handling
	FIA_IMA.1(Extension)	TOE internal mutual authentication
	FIA_SOS.1	Verification of confidential information

	FIA_UAU.2	Authenticate the user before every action
	FIA_UAU.4	Anti-reuse authentication mechanism
	FIA_UAU.7	Certification Feedback Protection
	FIA_UID.2	Identify the user before every action
Security Management (FMT)	FMT_MOF.1	Security Function Management
	FMT_MTD.1	TSF Data Management
	FMT_PWD.1(Extension)	Identity & Password Management
	FMT_SMF.1	Management Function Specification
	FMT_SMR.1	Security Role
TSF Protection (FPT)	FPT_ITT.1	Basic protection of internally transmitted TSF data
	FPT_PST.1(Extension)	Basic protection of stored TSF data
	FPT_TST.1	TSF Self-Examination
TOE Access (FTA)	FTA_MCS.2	Limit the number of concurrent sessions per user property
	FTA_SSL.3	Session termination by TSF
	FTA_TSE.1	Setting Up a TOE Session
Safe Paths/Channels (FTP)	FTP_ITC.1	Secure channels between TSFs

[Table 5-1] Security Feature Requirements (SFR)

5.1.1. Security Audit (FAU)

FAU_ARP.1 Security alarms

Hierarchical to: No other components

Dependencies: FAU_SAA.1 Potential violation analysis

FAU_ARP.1.1 The TSF shall [send an email alert to authorized administrators] upon detection of a potential security violation.

FAU_GEN.1 Generating Audit Data

Hierarchy: None

Dependencies: FPT_STM.1 Reliable time stamps

FAU_GEN.1.1 The TSF must be able to create audit records of the following audited events:

a) Start-up and shutdown of the audit functions
b) *Not specified* : All audited incidents based on audit level
c) [[Table 5-2]'s "Audited Cases"]

FAU_GEN.1.2 At a minimum, the TSF must record the following information within each audit record:

a) Date and time of the event, type of incident, identity of the subject (if available), outcome of the event (success or failure)
b) For each type of audit incident, the "Audited Incident" in [Table 5-2]] is based on the Auditable Incident Definition of the Functional Component included in the Protection Profile/Security Objective Specification.

Security Component	Auditable Event	Additional Audit Information
FDP_UDE.1	User Data Cancer·Decryption successes and failures	
Identification & Authentication	Logging in and out of users	
	Registering, changing, or deleting users	
	Responding actions when the limit of user authentication attempts is reached	
	Any changes to passwords	
	Authentication failed due to credential reuse attempt detection	
Security Management	IP registration, deletion, and modification of the management device	
	Perform security management functions and change or delete all security attribute values.	Changed security attribute data
	Primary Account (ID)·Change your password	
	IP blocking for management terminal access	
Secure Session Management	Lock or terminate a user's session	
	What to do when multiple login attempts from the same account are detected	
	Deny new sessions based on concurrent session limit	
Generate a passkey	Failed to generate a passkey	

Use a password	Cryptographic failures (including password operation types)	
Audit Records	Starting and Ending the TOE Audit Function in Software	
	Response action in case of failure to save audit records	
Self-protection	Conduct your own exams	Failed security features
	Perform integrity verification of the TOE itself	Components for which integrity checks failed

[Table 5-2] Audited Cases

FAU_SAA.1 Potential Violation Analysis

Hierarchical to: No other components

Dependencies: FAU_GEN.1 Audit data generation

FAU_SAA.1.1 TSF must be able to enforce a set of rules when examining audited events, and this

You should be able to point out potential violations of the TSF on a rule-based basis.

FAU_SAA.1.2 The TSF shall apply the following rules when examining audited incidents:

a) Known [Among the audited cases in FIA_UAU.2, the audit case of failure of certification, Among the audited incidents in FPT_TST.1, audit incidents of integrity violations and failure of self-examinations; Failure of self-test of verified cryptographic module, Responding actions in anticipation of audit data loss as specified in FAU_STG.3, Authentication failures beyond 5 consecutive logins as specified in FIA_AFL.1; concurrent attempts to connect administrators of the same account as specified in FTA_MCS.2; Attempts to access security management from unregistered IPs specified in FTA_TSE.1 and Simultaneous attempts to access administrators with the same privileges accumulation or combination of]

b) [None]

FAU_SAR.1 Audit Review

Hierarchical to : No other components

Dependencies : FAU_GEN.1 Audit data generation

FAU_SAR.1.1 TSF [Authorized Administrator]From the audit record to [All audit data]should be provided with the ability to read.

FAU_SAR.1.2 TSF **Audit records must be provided to authorized administrators so that they can interpret the information.**

FAU_SAR.3 (1) Selectable Audit Review (API Audit Log)

Hierarchical to : No other components

Dependencies : FAU_SAR.1 Audit review

FAU_SAR.3.1 TSF stands for [AND] for audit data based on [Encryption policy ID, Crypto accounts, types of actions, results at Optionally review accordingly]It should provide the ability to apply the

FAU_SAR.3 (2) Selectable Audit Review (Administrator Audit Log)

Hierarchical to : No other components

Dependencies : FAU_SAR.1 Audit review

FAU_SAR.3.1 The TSF shall provide the ability to apply [selected reviews based on the task type, task operation, task type ID, operator, outcome, query period, and client IP] of audit data based on [AND].

FAU_STG.3 Action in case of possible audit data loss

Hierarchical to : No other components

Dependencies : FAU_STG.1 Protected audit trail storage

FAU_STG.3.1 The TSF should take the [notify authorized administrator, [none]] action if the audit trail exceeds [threshold above 80% of tablespace size].

FAU_STG.4 prevention of audit data loss

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

Dependencies: FAU_STG.1 Protected audit trail storage

FAU_STG.4.1 The TSF should *ignore the audited events* and execute [send alert mail to authorized administrator] if the audit repository is saturated.

5.1.2. Cryptographic Support (FCS)

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

Hierarchical to : No other components
Dependencies : FCS_COP.1 Cryptographic operation
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [algorithms described in [Table 5-3]] and the specified cryptographic key sizes [key sizes described in [Table 5-3]] that meet the following: [standards described in [Table 5-3]].

Classification	Cryptographic Key	List of Standards	Cryptographic Key Generation Algorithm	Cryptographic Key Size
User data encryption	User data encryption key	ISO/IEC 18031	HASH_DRBG (SHA256)	128bit
				192bit
				256bit

[Table 5-3] Encryption Key Generation Algorithm

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

Hierarchical to : No other components
Dependencies : FCS_COP.1 Cryptographic operation
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [algorithms described in [Table 5-4]] and the specified cryptographic key sizes [key sizes described in [Table 5-4]] that meet the following: [standards described in [Table 5-4]].

Classification	Cryptographic Key	List of Standards	Cryptographic Key Generation Algorithm	Cryptographic Key Size
TSF Data Encryption	Cryptographic key for the "User	ISO/IEC 18031	HASH_DRBG(SHA 256)	128bit

	data encryption key" (Master key)			
	Cryptographic key for the "Master key"	ISO/IEC 18031	HASH_DRBG(SHA 256)	128bit
	TSF data (Environment configuration file) Cryptographic key	ISO/IEC 18031	HASH_DRBG(SHA 256)	128bit
	Cryptographic key for data transfer (Session key)	ISO/IEC 18031	HASH_DRBG(SHA 256)	128bit
Mutual Authenticat ion	API Private Key	ISO/IEC 18033-2	RSAES(SHA-256)	2048 bit
	API Public Key	ISO/IEC 18033-2	RSAES(SHA-256)	2048 bit
	Policy Server Private Key	ISO/IEC 18033-2	RSAES(SHA-256)	2048 bit
	Policy Server Public Key	ISO/IEC 18033-2	RSAES(SHA-256)	2048 bit

[Table 5-4] Cryptographic key generation algorithm

FCS_CKM.4 Cryptographic key destruction

Hierarchical to : No other components
Dependencies : FCS_CKM.1 Cryptographic key generation

FCS_CKM.4.1 The TSF shall ensure that cryptographic keys are destroyed in accordance with a specified cryptographic key destruction method [key destruction methods described in [Table 5-5]] that meets the following: [None].

Cryptographic Key	Timing of Deletion	Cryptographic Key Destruction Method
User Data Encryption Key	API : At the end of the decryption Policy Server : "Scrap" in the "Manage encryption keys" menu When Performing Buttons	API : 0x00 encryption key information Encryption key through overwriting Destroyed. Policy Server : Delete from the keybox
Encryption key (master key) of "User Data Encryption Key"	Policy Server : "Scrap" in the "Manage encryption keys" menu When Performing Buttons	Delete from the keystore
Encryption key for data in transit (Session Key):	At the end of the transmission data encryption/decryption	0x00 encryption key information Encryption key through overwriting Destroyed.

[Table 5-5] Destruction of encryption key

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

Hierarchical to : No other components
Dependencies : FCS_CKM.1 Cryptographic key generation
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1 The TSF shall perform [encryption and decryption of the user data] in accordance with a specified cryptographic algorithm [algorithms described in [Table 5-6]] and the cryptographic key sizes [key sizes described in [Table 5-6] that meet the following: [standards described in [Table 5-6]].

Cryptography	standard	Cryptographic algorithms	Encryption key length
Block Cryptography	KS X 1213	ARIA128	128bit
		ARIA192	192bit
		ARIA256	256bit
	TTAS. KO-12.004/R1	SEED	128bit
	ISO/IEC_10118-3	SHA224	224bit

hash		SHA256	256 bit
		SHA384	384 bit
		SHA512	512 bit

[Table 5-6] Cryptographic Algorithms (Encrypting User Data)

FCS_COP.1(2) Cryptographic operation (TSF data encryption/Mutual authentication)

Hierarchical to : No other components
Dependencies : FCS_CKM.1 Cryptographic key generation
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1 TSF has the following [[Specification of cryptographic algorithms that conform to the standard in Table 5-7] [[ticket 5-7]The cryptographic algorithm of] and the specified encryption key length [[ticket 5-7]The encryption key length depends on [[Table 5-7]of Cryptography]You have to do it.

Classification	Cryptographic Operation	Standards	Cryptographic Algorithm	Cryptographic Key Size
TSF Data Encryption	Block Cryptography	KS X 1213	ARIA128	128bit
	hash	ISO/IEC_10118-3	SHA256	256 bit
Mutual Authentication	Block Cryptography	KS X 1213	ARIA128	128bit
	hash	ISO/IEC_10118-3	SHA256	256 bit

[Table 5-7] Cryptographic Algorithms

FCS_RGB.1 Random bit generation (Extended)

Hierarchical to : No other components
Dependencies : No dependencies

FCS_RGB.1.1 The TSF generates the random bits required for generating cryptographic keys using a specified random bit generator that meets the following: [ISO/IEC 18031].

5.1.3. User Data Protection (FDP)

FDP_UDE.1 User data encryption

Hierarchical to : No other components
Dependencies : FCS_COP.1 Cryptographic operation

FDP_UDE.1.1 The TSF shall provide the TOE user with the capability to encrypt/decrypt user data in accordance with a specified [encryption/decryption method by column, [None]].

FDP_RIP.1 Subset residual information protection

Hierarchical to : No other components
Dependencies : No dependencies

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

5.1.4. Identification & Authentication (FIA)

FIA_AFL.1 Authentication failure handling

Hierarchical to : No other components
Dependencies : FIA_UAU.1 Authentication

FIA_AFL.1.1 The TSF shall detect when *[5]* unsuccessful authentication attempts occur in relation to [administrator authentication].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been *surpassed*, the TSF shall [send an email to the administrators and lock the account for 5 minutes].

FIA_IMA.1 TOE internal mutual authentication

Hierarchical to : No other components
Dependencies : No dependencies

FIA_IMA.1.1 The TSF shall perform mutual authentication using a [self-implemented authentication protocol] between [APIAgent and PolicyServer, PluginAgent and Policy Server] in accordance with [none].

FIA_SOS.1 The TSF shall provide a mechanism for verifying that secrets meet the [9 to 20 characters with a combination of uppercase and lowercase letters, numbers, and special characters] requirement.

category	content
Code of Conduct	Length of more than 9 digits
	Contains at least one number, uppercase letter, lowercase letter, and special character
Prohibited Items	Don't set the same password as your user account (ID)
	Same letter·Prohibition of continuous repetition of numbers
	Prohibit sequential input of consecutive letters or numbers on the keyboard
	Do not reuse previously used passwords

[Table 5-8] Password Security Standards

FIA_UAU.2 User authentication before any action

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

FIA_UAU.2.1 The TSF shall require each authorized administrator to be authenticated successfully before allowing any other TSF-mediated actions on behalf of such authorized administrator.

FIA_UAU.4 Anti-Reuse Authentication Mechanisms

Hierarchical to : No other components
Dependencies : FIA_UID.1 Identification

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

FIA_UAU.7 Protected authentication feedback

Hierarchical to : No other components
Dependencies : FIA_UID.1 Identification

FIA_UAU.7.1 The TSF shall provide only a/an [password masked (*) during input, authentication failure message when authentication failed] to the user while the authentication is in progress.

FIA_UID.2 User identification before any action

Hierarchical to : FIA_UID.1 Identification
Dependencies : No dependencies

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

5.1.5. Security Management (FMT)

FMT_MOF.1 Management of security functions behavior

Hierarchical to : No other components
Dependencies : FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

FMT_MOF.1.1 The TSF shall restrict the ability to ***enable the management behavior of*** the functions [list of security functions in [Table 5-8]] to [authorized administrators].

Subcategories	Care features	Managing Entity
Identification & Authentication	Registering, deleting, modifying, and authorizing users	Authorized Administrator
Security Management	Registering, deleting, and modifying the IP of the management device	Authorized Administrator
	Security Policy Management - Policy Settings	Authorized Administrator
	Encryption key management	Authorized Administrator
	API Management	Authorized Administrator
Self-protection	Perform integrity checks	Authorized Administrator
Update Protection	TOE Version Information Lookup	Authorized Administrator
Audit Records	Inquiry of audit records	Authorized

		Administrator

[Table 5-9] List of Security Functions Management

FMT_MTD.1TSF Management of TSF data

Hierarchical to : No other components

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

FMT_MTD.1.1 The TSF shall restrict the ability to *control* the [data described in Table 5-9]to [authorized administrators].

List of TSF data	administration
Administrator's password	modification
API Citizen Information	inquiry
	new
	modification
	delete
Allow admin login IPs	inquiry
	new
	modification
	delete
About Encryption Key Groups	inquiry
	new
	modification
	delete
About encryption keys	inquiry
	new
	modification
	delete

About Encryption Rules	inquiry
	new
	modification
	delete
About Encryption Policies	inquiry
	new
	modification
	delete
API Installation IP	inquiry
	new
	delete
The encryption key of the "master key"	Generate
"Configuration File" Encryption Key	Generate
Key pairs for mutual authentication	Generate

[Table 5-10] TSF Data Management List

FMT_PWD.1 Management of ID and password (Extended)

Hierarchical to : No other components
Dependencies : FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

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

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

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

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: [
Items specified in FMT_MOF.1 Management of security functions behavior,
Items specified in FMT_MTD.1 Management of TSF data,
Items specified in FMT_PWD.1 Management of ID and password (Extended)
]

FMT_SMR.1 Security roles

Hierarchical to: No other components
Dependencies: FIA_UID.1 Identification

FMT_SMR.1.1 The TSF shall maintain the **[*administrator*]** role.
FMT_SMR.1.2 The TSF shall be able to associate users with the **role specified in FMT_SMR.1.1**.

5.1.6. Protection of the TSF (FPT)

FPT_ITT.1 Basic internal TSF data transfer protection

Hierarchical to : No other components
Dependencies : No dependencies

FPT_ITT.1.1 The TSF shall protect TSF data through **encryption and message integrity verification** from *disclosure, modification* when it is transmitted between separate parts of the TOE.

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 [
a) administrator's password
b) cryptographic key
c) critical security parameter
d) TOE configuration value (Security Policy, Environment Configuration Parameter)
] stored in containers controlled by the TSF from unauthorized *disclosure, modification*.

FPT_TST.1 TSF Testing

Hierarchical to : No other components
Dependencies : No dependencies

FPT_TST.1.1 The TSF shall run self-tests *at start-up and periodically during normal operation* to demonstrate the correct operation of the *TSF*.

FPT_TST.1.2 The TSF shall provide **authorized users** with the capability to verify the integrity of *[the TSF data in [Table 5-10]]*.

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

category	type	name	explanation
DBAPI_ Module	Library Files	xecuredbapi.jar	Decryption Library (Java)
		libxecuredbapi.so	Encryption/decryption library (C/C++)
		libxdbdbapi_comm.so	Oracle DB API library

		libXecureASN.so	Modules for ASN.1 Operations
		libXecureCSP.so	A module that provides an interface between a password library and other modules
		libXecureCodec.so	A module that provides a function for converting strings
		libXecureIO.so	Modules that provide functions related to memory, files, sockets, time, etc.
		libXecurePKCS5.so	Password-based encryption
		libXecurePKCS8.so	A module that controls the user's secret key information
		libXecureTLS.so	Modules that provide functions required for SSL and TCPIP communication
		libXecureCrypto.so	Verifiable Cryptographic Module Interface Library
	Config file	xdf.ini	DB API configuration file
		config.json	DB API Encryption configuration file
	Mutual Authentication File	api-pri.key	API Private Key (Mutual Authentication)
		api-pub.key	API Public Key (Mutual Authentication)
		pol-pub.key	Policy Server public key (Mutual authentication)
		tsf-enc.key	DB API TSF Data Encryption Key

API_Module	Library Files	xecuredbapi.jar	Decryption Library (Java)
		libxecuredbapi.so	Encryption/decryption library (C/C++)
		libXecureASN.so	Modules for ASN.1 Operations
		libXecureCSP.so	A module that provides an interface between a password library and other modules
		libXecureCodec.so	A module that provides a function for converting strings
		libXecureIO.so	Modules that provide functions related to memory, files, sockets, time, etc.
		libXecurePKCS5.so	Password-based encryption
		libXecurePKCS8.so	A module that controls the user's secret key information
		libXecureTLS.so	Modules that provide functions required for SSL and TCPIP communication
		libXecureCrypto.so	Verifiable Cryptographic Module Interface Library
	Config file	xdf.ini	DB API configuration file
		config.json	DB API Encryption configuration file
	Mutual Authentication File	api-pri.key	API Private Key (Mutual Authentication)
		api-pub.key	API Public Key (Mutual Authentication)

		pol-pub.key	Policy Server public key (Mutual Authentication)
		tsf-enc.key	DB API TSF Data Encryption Key
Policy_Server	Integrity verification is performed on all files below Hancom_xDB_V5.0/ in the path where Policy Server is installed. (Excludes Hancom_xDB_V5.0/logs.)		

[Table 5-11] TSF Data

5.1.7. TOE Access (FTA)

FTA_MCS.2 Per user attribute limitation on multiple concurrent sessions

Hierarchical to : FTA_MCS.1 Basic limitation on multiple concurrent sessions
Dependencies : FIA_UID.1 Identification

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

FTA_MCS.2.2 The TSF shall enforce, by default, a limit of [1] session per **administrator**.

FTA_SSL.3 Session termination by TSF

Hierarchy: None
Dependencies: FIA_UAU.1 Authentication

FTA_SSL.3.1 The TSF shall *terminate an interactive session* after [5 minutes of administrator inactivity].

FTA_TSE.1 TOE session establishment

Hierarchical to : No other components
Dependencies : No dependencies

FTA_TSE.1.1 The TSF shall be able to deny establishment of an **administrator's administrative access session** based on [access IP, *whether or not another administrator account with the same privilege has already activated an administrative access session*].

5.1.8. Secure Path/Channel (FTP)

FTP_ITC.1 Secure Channels Between TSFs

Hierarchy: None

Dependencies: None

FTP_ITC.1.1 The TSF shall provide a communication channel that is logically distinct from other communication channels between itself and other trusted IT products, provides guaranteed identification of terminals, and protects channel data from alteration or exposure.

FTP_ITC.1.2 The TSF shall allow trusted IT products to initiate communications over secure channels.

FTP_ITC.1.3 TSF must initiate communication over a secure channel for [sending alert mail].

5.2. Security Assurance Requirements

The assurance requirements of this ST consist of assurance components in CC Part 3, and its evaluation assurance level is EAL1+. The following table summarizes the assurance components:

Assurance Class	Assurance Component	
Security Target Assessment	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	Operation Guide
	AGD_PRE.1	Preparative procedures
Life Cycle Support	ALC_CMC.1	Labeling of the TOE
	ALC_CMS.1	TOE CM coverage
Tests	ATE_FUN.1	Functional testing
	ATE_IND.1	Independent testing – conformance
Vulnerability Assessment	AVA_VAN.1	Vulnerability survey

[Table 5-12] Assurance Requirements

5.2.1. Security Target

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 element

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

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

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

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

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

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

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

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

Evaluator action elements

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

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

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 element

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

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

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

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

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

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

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

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

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

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

Evaluator action elements

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

ASE_OBJ.1 Security Objectives for the Operational Environment

Dependencies : No dependencies

Developer action elements

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

Content and presentation element

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

Evaluator action elements

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

ASE_ECD.1 Extended components definition

Dependencies : No dependencies

Developer action elements

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

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

Content and presentation element

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 so 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 the existing components.

ASE_REQ.1 Stated security requirements

Dependencies : ASE_ECD.1 Extended components definition

Developer action elements

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

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

Content and presentation element

ASE_REQ.1.1C The statement of security requirements shall describe the SFR and the SAR.

ASE_REQ.1.2C All subjects, objects, operations, security attributes, external entities, and other terms used in the SFR and the SAR shall be defined.

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

ASE_REQ.1.4C All operations shall be performed correctly.

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

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

Evaluator action elements

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

ASE_TSS.1 TOE summary specification

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

Developer action elements

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

Content and presentation element

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

Evaluator action elements

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

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

5.2.2. Development

ADV_FSP.1 Basic functional specification

Dependencies: No dependencies

Developer action elements

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

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

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 SFR.

Content and presentation element

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 the rationale for the implicit categorization of interfaces as SFR-non-interfering.

ADV_FSP.1.4C The tracing shall demonstrate that the SFR trace to TSFI 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, complete instantiation of the SFR.

5.2.3. Guidance documents

AGD_OPE.1 Operation Guide

Dependences : ADV_FSP.1 Basic functional specification

Developer action elements

AGD_OPE.1.1D The developer should provide the Operation Guide.

Content and presentation element

AGD_OPE.1.1C The Operation Guide should describe -- for each user role --the user-

accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2C The Operation Guide should describe -- for each user role -- how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3C The Operation Guide should describe -- for each user role -- the available functions and interfaces, particularly all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4C The Operation Guide should suggest -- 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 Operation Guide shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences, and the implications for maintaining secure operation.

AGD_OPE.1.6C The Operation Guide 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 Operation Guide shall be clear and reasonable.

Evaluator action elements

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

AGD_PRE.1 Preparative procedures

Dependencies : No dependencies

Developer action elements

AGD_PRE.1.1D The developer need to provide TOE including the preparative procedures.

Content and presentation element

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

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

the operational environment as described in the ST.

Evaluator action elements

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

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

5.2.4. Life-cycle Support

ALC_CMC.1 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 element

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

Evaluator action elements

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

ALC_CMS.1 TOE CM coverage

Dependencies : No dependencies

Developer action elements

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

Content and presentation element

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

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

Evaluator action elements

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

requirements for content and presentation of evidence.

5.2.5. Tests

ATE_FUN.1 Functional testing

Dependencies : ATE_COV.1 Evidence of coverage

Developer action elements

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

ATE_FUN.1.2D The developer shall provide test documentation.

Content and presentation element

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

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

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

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

Evaluator action elements

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

ATE_IND.1 Independent testing – conformance

Dependencies : ADV_FSP.1 Basic functional specification
AGD_OPE.1 Operation Guide
AGD_PRE.1 Preparative procedures

Developer action elements

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

Content and presentation element

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

Evaluator action elements

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

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

5.2.6. Vulnerability Assessment

AVA_VAN.1 Vulnerability survey

Dependencies : ADV_FSP.1 Basic functional specification
AGD_OPE.1 Operation Guide
AGD_PRE.1 Preparative procedures

Developer action elements

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

Content and presentation element

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

Evaluator action elements

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

AVA_VAN.2.2E The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.2.3E The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

5.3. Security Requirements Rationale

5.3.1. Dependency of the SFR

The following table shows the dependencies of the security feature requirements.

number	Security Components	Dependencies	Ref. No.
1	FAU_ARP.1	FAU_SAA.1	3
2	FAU_GEN.1	FPT_STM.1	Rationale (1)
3	FAU_SAA.1	FAU_GEN.1	2
4	FAU_SAR.1	FAU_GEN.1	2
5	FAU_SAR.3(1)	FAU_SAR.1	4
6	FAU_SAR.3(2)	FAU_SAR.1	4
7	FAU_STG.3	FAU_STG.1	Rationale (2)
8	FAU_STG.4	FAU_STG.1	Rationale (2)
9	FCS_CKM.1(1)	FCS_COP.1	12
		FCS_CKM.4	11
10	FCS_CKM.1(2)	FCS_COP.1	13
		FCS_CKM.4	11
11	FCS_CKM.4	FCS_CKM.1	9, 10
12	FCS_COP.1(1)	FCS_CKM.1	9
		FCS_CKM.4	11
13	FCS_COP.1(2)	FCS_CKM.1	10
		FCS_CKM.4	11
14	FCS_RBG.1	-	-
15	FDP_UDE.1	FCS_COP.1	12
16	FDP_RIP.1	-	-
17	FIA_AFL.1	FIA_UAU.1	20
18	FIA_IMA.1	-	-
19	FIA_SOS.1	-	-
20	FIA_UAU.2	FIA_UID.1	23
21	FIA_UAU.4	-	-
22	FIA_UAU.7	FIA_UAU.1	20
23	FIA_UID.2	-	-
24	FMT_MOF.1	FMT_SMF.1	27

		FMT_SMR.1	28
25	FMT_MTD.1	FMT_SMF.1	27
		FMT_SMR.1	28
26	FMT_PWD.1	FMT_SMF.1	27
		FMT_SMR.1	28
27	FMT_SMF.1	-	-
28	FMT_SMR.1	FIA_UID.1	23
29	FPT_ITT.1	-	-
30	FPT_PST.1	-	-
31	FPT_TST.1	-	-
32	FTA_MCS.2	FIA_UID.1	23
33	FTA_SSL.3	-	20
34	FTA_TSE.1	-	-
35	FTP_ITC.1	-	-

[Table 5-13] Dependencies Rationale

Rationale (1) : FAU_GEN.1 has a dependency on FPT_STM.1, and this dependency is satisfied because FPT_STM.1 is met by the OE.Time stamp, a security objective for the operational environment.

Rationale (2): FAU_STG.3 and FAU_STG.4 have a dependency on FAU_STG.1, and these dependencies are satisfied because FAU_STG.1 is met by OE.DBMS, a security objective for the operational environment.

5.3.2. Assurance Requirements Rationale

Since the dependency of the EAL1 assurance package provided by the CC is already satisfied, the rationale for this is omitted.

An additional assurance requirement, ATE_FUN.1, includes ATE_COV.1 as a dependency. ATE_FUN.1 has been added to confirm that the developer has correctly performed the tests with the test items and recorded the results in the test sheet. Note, however, that ATE_COV.1, which shows the consistency between the test items and TSFI, is not included in this ST as it is not deemed necessary.

6. TOE Summary Specification

This chapter describes how the TOE satisfies the SFR for the security functions of the TOE: Security Audit, Cryptographic Support, User Data Protection, Identification and Authentication, Security Management, Protection of the TSF, and TOE Access.

6.1 Security Audit

The TOE generates and stores audit data for security audit events that occur in each TOE component. The audit data includes the date and time of event, type of event, identity of the subject, and outcome of the event. If a potential security violation is detected, the user will be notified in real time, and the incident will be dealt with in a manner specified according to the type of violation.

6.1.1. Security Alarms

Upon detecting a potential security violation, the TOE performs countermeasures against the security violation activity. The potential security violations and its countermeasures are shown in [Table 6-3].

6.1.2. Audit Data Generation

The TOE creates audit records for the auditable events in [Table 6-2], including the operation of the security functions provided by the TOE and the history of security management. The information that is recorded when the audit record is generated includes the event occurrence time of the audit target, subject information such as item, ID, or access IP, and processing results. As a policy server and security management server of the TOE, Policy Server generates audit data for the auditable events defined in FAU_GEN.1, including user data encryption/decryption performed by APIAgent and PluginAgent, and stores them in the DBMS.

Division	Field	Details
APIAgent/PluginAgent audit log (Audit data on security functions performed)	Log timestamp, task type, task operation, result, encryption policy ID, API user ID,	Audit records related to the security functions provided by the TOE, such as user data encryption and decryption

	CLIENT, programs, log messages	
Administrator audit log (Security management behaviors and TSF management data)	Date, operator, task type, task operation, result, CLIENT IP	Audit records that record the results of additions, changes, and deletions to the TSF data by authorized administrators as well as management behaviors including authorized administrators' login to security management interfaces

[Table 6-1] Audit data created by the TOE

Security Component	Audit Incident	Additional Audit Information
FDP_UDE.1	User Data Cancer·Decryption successes and failures	
Identification & Authentication	Logging in and out of users	
	Registering, changing, or deleting users	
	Responding actions when the limit of user authentication attempts is reached	
	Any changes to passwords	
Security Management	IP registration, deletion, and modification of the management device	
	Perform security management functions and change or delete all security attribute values.	Changed security attribute data
	Primary Account (ID)·Change your password	
	IP blocking for management terminal access	
Secure Session Management	Lock or terminate a user's session	
	What to do when multiple login attempts from the same account are detected	
	Deny new sessions based on concurrent session limit	

Generate a passkey	Failed to generate a passkey	
Use a password	Cryptographic failures (including password operation types)	
Audit Records	Starting and Ending the TOE Audit Function in Software	
	Response action in case of failure to save audit records	
Self-protection	Conduct your own exams	Failed security features
	Perform integrity verification of the TOE itself	Components for which integrity checks failed

[Table 6-2] Cases subject to audit

6.1.3. Analysis and Response to Potential Violations

The TOE analyzes potential violations through audit events and implements predefined countermeasures.

Audit of potential violations	rule	Response Actions
Authentication failure audit event among auditable events under FIA_UAU.2	When accumulating one audit incident	Notify authorized administrators by email
Integrity Violations in Audited Incidents in FPT_TST.1 Audit case and self-test of verified cryptographic module Failure Incident	When accumulating one audit incident	Notify authorized administrators by email
Audit log data loss prediction as specified in FAU_STG.3	In the event of an audit incident	Notify authorized administrators by email
Logins specified in FIA_AFL.1 Authentication failed more than 5 times in a row	When accumulating one audit incident	Notify authorized administrators by email
Concurrent attempts to connect to administrators of the same account as specified in FTA_MCS.2	When accumulating one audit incident	Notify authorized administrators by email

Unregistered as specified in FTA_TSE.1 Attempts to access security management from IP and the same Attempts to access the administrator at the same time	When accumulating one audit incident	Notify authorized administrators by email

[Table 6-3] Audit Cases of Potential Violations and Response Actions

6.1.4. Audit Review and Selectable Audit Review

The TOE provides a function of reviewing and selectively reviewing the audit data generated by the TOE and stored in the DBMS. The stored API audit logs will be selectively reviewed according to query date, encryption policy ID, encryption account, operation type, and result. The administrator audit log will also be selectively reviewed according to the task type, task operation, operator, results, query period, and CLIENT IP. Authorized administrators can review and selectively review each audit log.

6.1.5. Audit: Predicting data loss, reacting behaviors and loss prevention

The audit records generated by the TOE are stored in a storage (DBMS) provided in the TOE operating environment. Only an authorized administrator can access the audit record DB through the storage and perform audit record cleanup tasks.

The TOE periodically checks the space of the audit record storage and, if it exceeds the remaining space required set by the authorized administrator, creates an audit record for the excess event and alerts the authorized administrator (by sending an alert email). When the audit record storage is full, the TOE ignores the audit details and alerts the authorized administrator (by sending an alert email) to protect the audit record.

- The default threshold for an excess alert is 80% of the total audit record storage capacity (based on the Tablespace), and it cannot be changed. When the threshold is exceeded, the authorized administrator is alerted (by sending an alert email).
- The default threshold for a full storage alert is 90% of the total audit record storage capacity (based on the Tablespace), and it cannot be changed. When the threshold is exceeded, the authorized administrator is alerted (by sending an alert email).

※ Related security functional requirements

- FAU_ARP.1, FAU_GEN.1, FAU_SAA.1, FAU_SAR.1, FAU_SAR.3, FAU_STG.3, FAU_STG.4

6.2. Cryptographic Support

6.2.1. Cryptographic Key Generation

The encryption of user data and TSF data uses symmetric key cryptography, and the cryptographic key required is generated with the HASH_DRBG algorithm that conforms to the ISO/IEC 18031 standard.

For the cryptographic key required for the asymmetric key cryptography, a 2048-bit cryptographic key is generated through the RSAES algorithm that conforms to the ISO / IEC 18033-2 (2006) standard.

Cryptographic keys managed by Hancom xDB V5.0 Policy Server are encrypted with the ARIA-CBC algorithm and stored and managed in the DBMS, and integrity verification is performed with the SHA256 algorithm.

The master key used to encrypt the cryptographic key is encrypted through ARIA-CBC (128bit) and managed in the DBMS, and only authorized users can access and change it.

The TOE performs cryptographic key generation using the cryptographic algorithm of the validated cryptographic module "XecureCrypto v2.1.0.0" in [Table 6-5] whose safety and implementation conformity have been verified by the Korea Cryptographic Module Validation Program (KCMVP). The cryptographic algorithm and cryptographic key size for each cryptographic key are shown in [Table 6-4] below.

use	Passkey	List of Standards	Encryption Key Generation Algorithm	Passkey length
User Data Encryption	User Data Encryption Key	ISO/IEC 18031	Hash_DRBG (SHA256)	128bit
				192bit
				256bit
TSF Data Encryption	Encryption key (master key) of "User Data Encryption Key"	ISO/IEC 18031	Hash_DRBG (SHA256)	128bit
	The encryption key of the "master key"	Strength. CO-12.0334	PBKDF(HMAC-SHA256)	128bit
	API configuration files Encryption key	ISO/IEC 18031	Hash_DRBG (SHA256)	128bit
	Encryption key for data in transit	ISO/IEC 18031	Hash_DRBG (SHA256)	128bit

	(Session Key)			
Mutual Authentication	API Private Key	ISO/IEC 18033-2	RSAES (SHA256)	2048bit
	API Public Key	ISO/IEC 18033-2	RSAES (SHA256)	2048bit
	Policy Server Private Key	ISO/IEC 18033-2	RSAES (SHA256)	2048bit
	Policy Server Public Key	ISO/IEC 18033-2	RSAES (SHA256)	2048bit

[Table 6-4] Cryptographic keys and their generation algorithms

category	content
Cryptographic Module Name	XecureCrypto v2.1.0.0
Verification number	CM-247-2029.5
Verification Grade	VSL1
Developer	HANCOM WITH
Verification date	May 30, 2024

[Table 6-5] General information on the validated cryptographic module

6.2.2. Cryptographic Operation

The TOE performs cryptographic operation for user data encryption using the cryptographic algorithm of the validated cryptographic module "XecureCrypto v2.0.1.1" whose safety and implementation conformity have been verified by the Korea Cryptographic Module Validation Program (KCMVP). During cryptographic operation, the validated cryptographic module is operated only as verification object in cryptographic module verification standard. When performing encryption using the block cipher algorithm, the ECB mode is not used regardless of the size of the plain text. For the use of IV in CBC, CFB, and OFB modes and the use of a counter in CTR mode, the methods provided in KS X 1212 and TTAS.KO-12.004/R1 will be applied. The standards, cryptographic algorithms, and cryptographic key sizes used in cryptographic operation when encrypting user data are shown in [Table 6-6].

Cryptography	standard	Cryptographic algorithms	Encryption key length
Block Cryptography	KS X 1213	ARIA128	128 bits
		ARIA192	192 bits

		ARIA256	256 bits
	TTAS. KO-12.004/R1	SEED	128 bits
hash	ISO/IEC_10118-3	SHA224	224bits
		SHA256	256 bits
		SHA384	384 bits
		SHA512	512 bits

[Table 6-6] Cryptographic operation algorithm (User data encryption)

The TOE performs cryptographic operation for TSF data encryption using the cryptographic algorithm of the validated cryptographic module "XecureCrypto v2.0.1.1" whose safety and implementation conformity have been verified by the Korea Cryptographic Module Validation Program (KCMVP). During cryptographic operation, the validated cryptographic module is operated only as verification object in cryptographic module verification standard. When performing encryption using the block cipher algorithm, the ECB mode is not used regardless of the size of the plain text. For the use of IV in CBC, CFB, and OFB modes and the use of a counter in CTR mode, the method provided in KS X 1212 and TTAS.KO-12.004/R1 will be applied. The standards, cryptographic algorithms, and cryptographic key sizes used in cryptographic operation when encrypting TSF data are shown in [Table 6-7].

Cryptographic Operation	Standards	Cryptographic Algorithm	Cryptographic Key Size
Block Cryptography	KS X 1213	ARIA128	128 bits
hash	ISO/IEC_10118-3	SHA256	256 bits

[Table 6-7] Cryptographic operation algorithm (TSF data encryption)

In order to perform mutual authentication between the Policy Server and the API, TOE performs cryptographic operations using the cryptographic algorithm subject to verification of the cryptographic module 'XecureCrypto v2.1.0.0', whose safety and suitability have been confirmed through the Cryptographic Module Verification System (KCMVP). When performing cryptographic operations, the cryptographic module that must be verified operates in the operation mode subject to verification. Policy Server and API each have a pair of private and public keys issued before installation, and mutual authentication protocols are performed when connecting to a session.

The API generates a signature value by signing a specific message with the API private key. Policy Server receives the signature sent by the API, verifies the signature with the API public

key, generates a specific message, signs it with the private key of Policy Server, and generates the signature value generated by Policy Server. After receiving this value, the API performs mutual authentication by verifying the signature with the public key of the Policy Server. The standard and cryptographic algorithm key lengths used for mutual authentication are shown in Table 6-8.

Cryptography	standard	Cryptographic algorithms	Encryption key length
Public Key Cryptography	ISO/IEC 18033-2	RSAES(SHA-256)	2048 bit
Block Cryptography	KS X 1213	ARIA128	128bit
hash	ISO/IEC_10118-3	SHA256	256 bit

[Table 6-8] Algorithm for Cryptography (Mutual Authentication)

6.2.3. Cryptographic Key Destruction

The cryptographic key loaded in memory during generation, distribution, and operation of the key will be destroyed by overwriting all random bits with 0x00 after its validity period.

Passkey	When are they deleted?	How to destroy the encryption key
User Data Encryption Key	DB API : At the end of the cancer/decryption operation Policy Server : "Scrap" in the "Manage encryption keys" menu When Performing Buttons	DB API: Destroy the encryption key by overwriting the encryption key information with 0x00. Policy Server : Delete from the keybox
Encryption key (master key) of "User Data Encryption Key"	Policy Server : "Scrap" in the "Manage encryption keys" menu When Performing Buttons	Delete from the keystore
Encryption key for data in transit (Session Key)	At the end of the transmission data encryption/decryption	The encryption key is destroyed by overwriting the encryption key information with 0x00.

[Table 6-9] Destruction of the encryption key

6.2.4. Random Number Generation

TOE generates random numbers required for generating encryption keys using a HASH_DRBG (256-bit) algorithm through the random number generator of 'XecureCrypto v2.1.0.0', a cryptographic module that has been verified for safety and suitability for implementation through the cryptographic module verification system.
General information on the validated cryptographic module is provided in [Table 6-5].

※ Related security functional requirements

- FCS_CKM.1, FCS_CKM.4, FCS_COP.1, FCS_RBG.1

6.3. User Data Protection

6.3.1. User Data Encryption

The TOE provides the function of encrypting and decrypting user data by column. In addition, in order to prevent the same cipher text from being generated for the same plain text data when encrypting user data, a random initial vector (IV) is used for encryption.

6.3.2. Subset residual information protection

When encrypting user data using an API method, the application developer shall protect user data by allowing original data to be deleted after user data encryption/decryption.

※ Related security functional requirements

FDP_UDE.1,FDP_RIP.1

6.4. Identification & Authentication

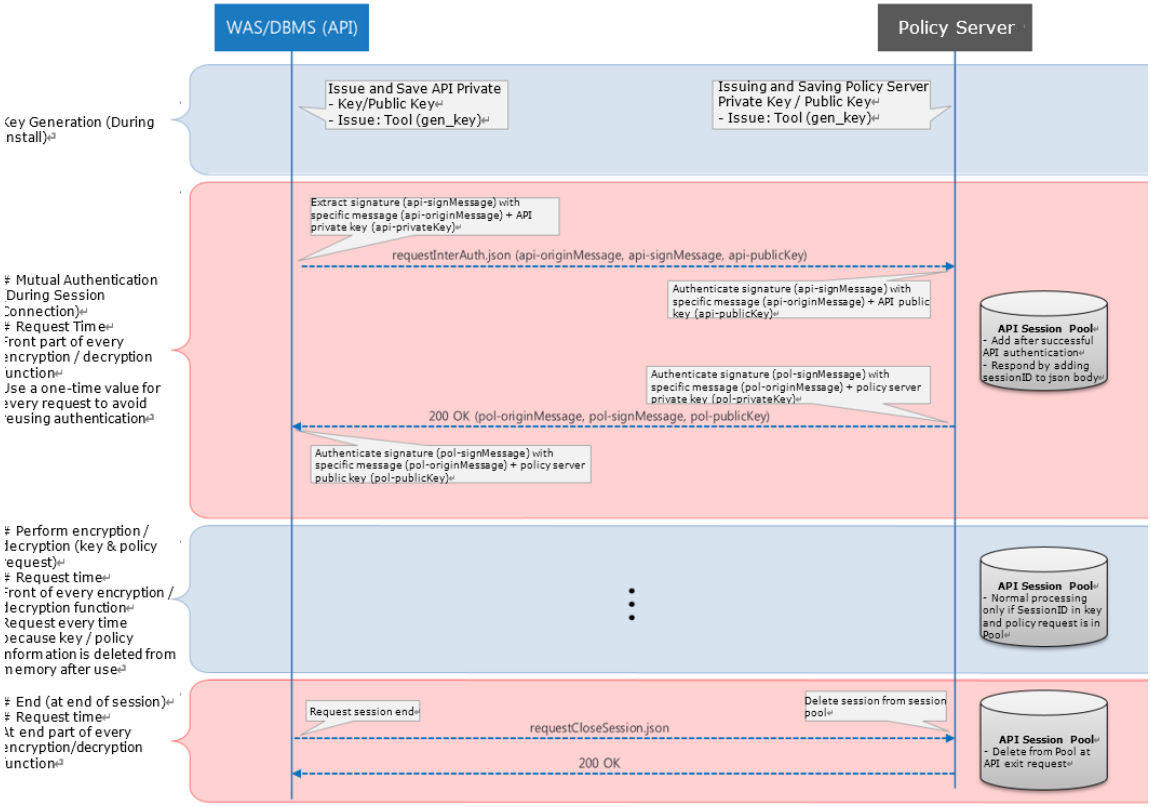
The TOE satisfies the identification and authentication of an authorized administrator through the verification of passwords.

6.4.1. Handling authentication failures

If authentication fails 5 times in a row when identifying and authenticating an authorized administrator's administrative access, the management access attempt of the account will be blocked for 5 minutes and an administrator email notification will be sent. It also stores audit records of authentication failures.

6.4.2. Mutual Authentication

Mutual authentication between API and Policy_Server is performed through a self-implemented authentication protocol. The mutual authentication method is shown in Figure 6-1.



[Figure 6-1] Mutual authentication flow between API, DBAPI, and Policy Server

6.4.3. Password Policy Validation

Validate the password value according to the password combination rules set when creating and changing the password of the security administrator.
When creating a password, the following verification mechanism is provided.

category	content
Code of Conduct	9 ~ 20 digits or less
	Contains at least one number, uppercase letter, lowercase letter, and special character - Uppercase letters: A–Z (26) - Lowercase letters: a–z (26) - Numbers: 0–9 (10) - Special characters:!, @, #,%, ^,*, (,), -, =, _, +, [,], {, }; ,., /, >, ?
Prohibited Items	Don't set the same password as your user account (ID)
	Same letter·Prohibition of continuous repetition of numbers
	Prohibit sequential input of consecutive letters or numbers on the keyboard
	Do not reuse previously used passwords

[Table 6-10] Password Security Standards

6.4.4. Identification & Authentication

The TOE shall require authorized administrators to be authenticated successfully before allowing them to access and control all security functions. The TOE provides a password-based authentication mechanism that enables identification and authentication using the ID and password.

6.4.5. Single-use Authentication Mechanisms

The TOE prevents reuse of authentication data by using random bit during authentication.
The TOE generates a hash value by combining the login ID, login password, and random bit (TOKEN) during authentication.
At this time, random number generation and hash value generation are generated using the validated cryptographic module "XecureCrypto v.2.1.0.0" . The generated hash value is managed in memory after login, and the integrity value of the authentication information is checked at the time of login, preventing the reuse of the authentication information.

6.4.6. Protected Authentication Feedback

The TOE does not provide feedback on the reason for the failure when authentication fails, and the password inputted during authentication or when registering or changing the password is masked with the "*" character to prevent the password from being displayed on the screen.

※ Related security functional requirements

FIA_AFL.1, FIA_IMA.1, FIA_SOS.1, FIA_UAU.2, FIA_UAU.4, FIA_UAU.7, FIA_UID.2

6.5. Security Management

The security administrator must perform the security management function through the security management interface, Policy Server, and must go through the identification and authentication process to use the security management function. Security management functions can be divided into administrator account management, key management, policy management, and security management interface settings.

6.5.1. Management of Security Functions

Only after successful execution of self-enforced identification and authentication functions is the administrative access control function called, and the authorized administrator (administrator) is allowed to access the security management interface through a secure channel (SSL).

The TOE provides the authorized administrators (administrators) with the following security functions:

Subcategories	Care features	Managing Entity
Identification & Authentication	Registering, deleting, modifying, and authorizing users	Authorized Administrator
Security Management	Registering, deleting, and modifying the IP of the management device	Authorized Administrator
	Security Policy Management - Policy Settings	Authorized Administrator
	Encryption key management	Authorized Administrator
	API Management	Authorized Administrator
Self-protection	Perform integrity checks	Authorized Administrator
Update Protection	TOE Version Information Lookup	Authorized Administrator
Audit Records	Inquiry of audit records	Authorized Administrator

[Table 6-11] List of Security Features

6.5.2. Management of TSF Data

The TOE provides authorized administrators (administrators) with management functions for modifying, querying, deleting, and adding (creating) items on the TSF data list in [Table 6-12].

List of TSF data	administration
Administrator's password	modification
API	inquiry
	new
	modification
	delete
Allow admin login IPs	inquiry
	new
	modification
	delete
About Encryption Key Groups	inquiry
	new
	modification
	delete
About encryption keys	inquiry
	new
	modification
	delete
About Encryption Rules	inquiry
	new
	modification
	delete
About Encryption Policies	inquiry

	new
	modification
	delete
API Installation IP	inquiry
	new
	delete
The encryption key of the "master key"	Generate
"Configuration File" Encryption Key	Generate
Key pairs for mutual authentication	Generate

Table 6-12: TSF Data Management List

6.5.3. Management of ID and Password

Authorized administrators are forced to change the password upon their first login to the security management interface; the authorized administrator (administrator) can change the administrator's password through the security management interface.

6.5.4. Security Role

Security roles are not categorized, and only a single administrator role will be available. Security functions such as TSF data management, administrator account management, administrator profile and password rule management, DB encryption user management, administrator notification settings, administrator notification email management, monitoring, and management of allowed IPs for administrative access are limited to authorized administrators.

※ Related security functional requirements

- FMT_MOF.1, FMT_MTD.1, FMT_PWD.1, FMT_SMF.1, FMT_SMR.1

6.6. Protection of the TSF

6.6.1. Basic Internal TSF Data Transfer Protection

When the TSF data is transmitted between the separated parts of the TOE using the cryptographic target algorithm of the validated cryptographic module "XecureCrypto v.2.1.0.0" a validated cryptographic module whose safety and implementation conformity are verified through the cryptographic module verification system (KCMVP), it protects the transmitted TSF data such as audit data and important security parameters from exposure and modification.

1. The API module installs each private key (PKCS#8 standard) and public key generated by the Policy Server at the time of installation.
2. The API module generates a random value (Session Key: 16byte Random: HASH_DRBG) using the Verified Cryptographic Module (KCMVP) when transmitting TSF data to the Policy Server. The generated random value encrypts the TSF data (ARIA 128, CBC mode) and generates a checksum (SHA256) for the value before the TSF data encryption. Then, it encrypts the random value with the public key (RSA-OAEP, 2048) and sends a message (TSF data) that combines the random value, the TSF data encryption value, and the checksum value.
3. After receiving the request message (TSF data), the Policy Server decrypts the random value with the private key (PKCS#8 standard) (RSA-OAEP, 2048) and decrypts the request message (TSF data) with the decrypted random value (ARIA 128, CBC mode).
4. Verify the integrity of TSF data through checksum (SHA256) verification of decrypted TSF data and perform tasks for requests (mutual authentication, distribution of user data keys).
5. After completing the work, the Policy Server generates a random value (Session Key: 16byte Random: HASH_DRBG) using the Verification Encryption Module (KCMVP). Encrypt the response message (TSF data) with the generated random value (ARIA 128, CBC mode) and generate a checksum (SHA256) for the value before encryption of the response message (TSF data). Then, by encrypting the random value with the public key (RSA-OAEP, 2048), the response message (TSF data) is sent to the DB API by combining the random value, TSF data encryption value, and checksum value. .
6. If integrity verification fails, an error message is generated and a random value is generated using the verifiable cryptographic module (KCMVP) (Session Key: 16byte Random: HASH_DRBG). Encrypts the error message with the generated random value (ARIA 128, CBC mode) and generates a checksum (SHA256) for the value before the error message is encrypted. Then, by encrypting the random value with the public key (RSA-OAEP, 2048), the error message is sent to the DB API by combining the random value, TSF data encryption value, and checksum value. .

7. After receiving the response message, the DB API decrypts the random value with the private key (PKCS#8 standard) (RSA-OAEP, 2048) and decrypts the request message (TSF data) with the decrypted random value (ARIA 128, CBC mode).
8. Verify the integrity of TSF data through checksum (SHA256) verification of decrypted TSF data, and use the information received from Policy Server.

6.6.2. Basic protection of stored TSF data

TOE encrypts and stores and manages protected TSF data to protect it from unauthorized exposure and alteration of stored TSF data.
The information subject to mandatory encryption is the administrator password, TOE setting information (DB storage information and configuration file information), etc., and the administrator password is encrypted with SHA256 and the TOE setting value information is encrypted with ARIA-CBC 128bit.
The mandatory encryption target information among the Policy Server's Configuration file information is the Policy Server public key path, API public key path, API private key path, API private key password, API user ID, Policy Server IP information, and Policy Server Port information. The mandatory encryption target information managed in the Policy Server database is the super administrator password, master key, and user data encryption key information. The encryption target information within the API module's Configuration file information is the Policy Server public key path, API public key path, API private key path, API private key password, API user ID, Policy Server IP information, and Policy Server Port information.The list of protected TSF data and the applied cryptographic algorithms are as follows.

TSF Data		Application Algorithms and Data	Mandatory encryption targets
Administrator password		SHA256(Password+salt)	Mandatory encryption
TOE Setpoints	Policy Server public key path API Public Key Path API Private Key Path API Private Key Password API user ID Policy Server IP Info	ARIA-CBC(data)	Mandatory encryption

	Policy Server Port info		
Passkey	Encryption key (master key) of "User Data Encryption Key"	ARIA-CBC(key)	Mandatory encryption
	The encryption key of the "master key"	PBKDF2-HMAC-SHA256	-
	API/DB API Configuration File Encryption key	ARIA-CBC(key)	Mandatory encryption

[Table 6-13] Protected TSF data and applied cryptographic algorithms

6.6.3. Self-test

In order to demonstrate the correct operation of its component Policy Server, TOE periodically conducts its own tests at startup and during regular operation.

After verifying the integrity of the TSF data described below at start-up, the Policy Server retrieves its own PID value to check whether its process is working normally, and performs its own tests by checking the DBMS connection status and checking the operation of the mail server.

In addition, after conducting an integrity check periodically (1 day) during regular operation, the company conducts its own tests by checking its own PID value, checking the DBMS connection status, and checking the operation of the mail server.

If the test fails, create an audit log and send it by e-mail.

TOE provides the ability to verify the integrity of TSF data and TSF, including all files such as security policy files, main executable files, and configuration files required for TOE operation.

For integrity checks, TOE generates a Sing value for the integrity check objects and compares it with the stored Sign value (reference value), and the results of the self-test of the cryptographic module to be verified are included in the integrity test.

If a violation of integrity is found, the TOE notifies its authorized administrator and generates audit data. The TOE audits and records the results of its own tests, integrity checks, and the response actions of authorized administrators.

Integrity tests performed by the TOE can be divided into those that run on the Policy Server and those that run on the API.

Integrity checks in the Policy Server are operated at the time of operation, periodically, and at the request of the administrator, and can be summarized as the condition of occurrence of the integrity check in accordance with the TOE [Table 6-14]
When a violation occurs as a result of an integrity check, such as a compromise of integrity in the Policy Server, an audit log is generated and stored, and the authorized administrator is notified by e-mail. All integrity check files checked by the Policy Server are all except the log directory below Hancom_xDB_V5.0/.

The API and DBAPI modules perform integrity checks when calling the API, and when there is a violation as a result of the integrity check, such as compromising the integrity, the audit log is generated and sent to the Policy Server, which receives the audit log and sends it to the authorized administrator. For the integrity files of the API and DBAPI, see Table 6-16.

TOE Components	Conditions for Integrity Checks
Policy Server	at start-up, at the time desired by the authorized administrator during regular operation; Perform integrity checks periodically (1 day) during regular operation
Hancom xDB V5.0 API	Perform integrity checks on call
Hancom xDB V5.0 DBAPI	Perform integrity checks on call

[Table 6-14] Conditions for Integrity Checks to Occur According to TOE

kind	List of Standards	Encryption Key Generation Algorithm	Encryption key length
Integrity Verification	ISO/IEC 18033-2	RSAES (SHA256)	2048

[Table 6-15] Algorithms Used for Integrity Verification

TOE Type	type	name	explanation

Hancom xDB V5.0 API	Library Files	xecuredbapi.jar	Decryption Library (Java)
		libxecuredbapi.so	Encryption/decryption library (C/C++)
		libXecureASN.so	Modules for ASN.1 Operations
		libXecureCSP.so	A module that provides an interface between a cryptographic library and other modules.
		libXecureCodec.so	A module that provides a function for converting strings
		libXecureIO.so	Modules that provide functions related to memory, files, sockets, time, etc.
		libXecurePKCS5.so	Password-based encryption
		libXecurePKCS8.so	A module that controls the user's secret key information
		libXecureTLS.so	Modules that provide functions required for SSL and TCPIP communication
		libXecureCrypto.so	Verifiable Cryptographic Module Interface Library
	Config file	xdf.ini	DB API configuration file
		config.json	DB API Encryption configuration file
	Mutual Authentication File	api-pri.key	API Private Key (Mutual Authentication)
		api-pub.key	API Public Key (Mutual Authentication)
		pol-pub.key	Policy Server Public Key (Mutual Authentication)

		tsf-enc.key	DB API TSF Data Encryption Key
Hancom xDB V5.0 DBAPI	Library Files	xecuredbapi.jar	Decryption Library (Java)
		libxecuredbapi.so	Encryption/decryption library (C/C++)
		XDBAPIAgent.jar	DB APIAgent core library
		libxdbplugin_comm.so	Oracle DB API library
		libXecureASN.so	Modules for ASN.1 Operations
		libXecureCSP.so	A module that provides an interface between a cryptographic library and other modules.
		libXecureCodec.so	A module that provides a function for converting strings
		libXecureIO.so	Modules that provide functions related to memory, files, sockets, time, etc.
		libXecurePKCS5.so	Password-based encryption
		libXecurePKCS8.so	A module that controls the user's secret key information
		libXecureTLS.so	Modules that provide functions required for SSL and TCPIP communication
		libXecureCrypto.so	Verifiable Cryptographic Module Interface Library
	Config file	xdf.ini	DB API configuration file
		config.json	DB API Encryption configuration file

	Mutual Authentication File	api-pri.key	API Private Key (Mutual Authentication)
		api-pub.key	API Public Key (Mutual Authentication)
		pol-pub.key	Policy Server Public Key (Mutual Authentication)
		tsf-enc.key	DB API TSF Data Encryption Key
Hancom xDB V5.0 Policy Server	Integrity verification is performed on all files below Hancom_xDB_V5.0/ in the path where Policy Server is installed. (Excludes Hancom_xDB_V5.0/logs.)		

[Table 6-16] Integrity Verification Targets

※ Requirements for related security features

- FPT_ITT.1, FPT_PST.1, FPT_TST.1

6.7. TOE access

6.7.1. Per User Attribute Limitation on Multiple Concurrent Sessions

The TOE blocks the maximum number of concurrent sessions to 1 to disable concurrent login from the same account. The TOE also blocks concurrent logins with the same privilege. In the event of a concurrent login attempt from the same account or with the same privilege, it blocks the new login while maintaining the existing login.

6.7.2. Management of TSF-initiated Sessions (Extended)

The TOE terminates the session if there is no access to the administrator interface or action for a set period of time (5 minutes) after the authorized administrator's login.

6.7.3. TOE Session Management Settings

The TOE controls access to it so that only the registered IP (two default values or less) can access the security management interface. The allowed IPs can be set upon first login after the TOE is installed, and they can be added, changed, or deleted in security management using the allowed IPs list settings. IPs cannot be added by setting the IP address range; each IP address

must be added separately. In this case, settings that refer to the whole network range (0.0.0.0, 192.168.10. *, any, etc.) are not allowed.

※ Related security functional requirements

FTA_MCS.2, FTA_SSL.3, FTA_TSE.1

6.8. Safe Paths/Channels

6.8.1. Secure channels between TSFs

TOE supports TLS 1.2, a secure encrypted communication protocol, to protect transmitted data when sending alert mails.