Hancom xDB V2.8
Security Target(ST)
v1.10

HANCOM WITH

Document Revision History

Document Version	Notes	Date	Author(s)
v1.0	Registered a new document, Security Target: Hancom xDB V2.8	2018-03-16	Park Dong-gyu
v1.1	Revised the document to change the physical scope of the TOE	2019-04-02	Oh Byeong-jin
v1.2	Revised the document to reflect the validation extension of the cryptographic module	2019-05-14	Oh Byeong-jin
v1.3	Renamed the TOE	2019-05-20	Oh Byeong-jin
v1.4	Changed the operational environment (DBMS/Java version)	2019-06-03	Oh Byeong-jin
v1.5	Revised the document to reflect observations	2019-06-20	Oh Byeong-jin
v1.6	Updated the CI due to company name change Revised the document to use the same title on the cover as that in the reference Added rationale for conformance claims Revised the document to add a security objective for the operational environment "Trusted path/channels", etc.	2019-08-09	Oh Byeong-jin
v1.7	Changed TOE version (2.8.2.0 -> 2.8.2.2) Hancom xDB Policy Server Changed the package file's name	2019-08-26	Oh Byeong-jin
v1.8	Updated TSF data management section (TSF data descriptions revised) Updated the cryptographic key distribution section (Standard algorithm specified)	2019-09-06	Oh Byeong-jin
v1.9	Changed TOE version (2.8.2.2 -> 2.8.2.3)	2019-09-25	Oh Byeong-jin
v1.10	Changed ST dodument version (v1.9 -> v1.10)	2019-10-21	Oh Byeong-jin

Contents

1. Introduction to the Security Target

This document is the Security Target (ST) of Hancom xDB V2.8 (“TOE”). It defines the security functions and assurance requirements of the TOE, presents the definitions of the security issues, security objectives, IT security requirements, and TOE summary specifications, and discusses the rationale accordingly.

This Security Target is organized as follows:

- 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 V2.8
ST Version	v1.10
Author(s)	Oh Byeong-jin (Trust Platform Development Team, Hancom With Inc.)
Creation Date	October 21, 2019
Evaluation Criteria	Common Criteria for Information Security System (Notice of the Ministry of Science, ICT and Future Planning, No. 2016-73)
Common Criteria Version	CC V3.1 r5
Evaluation Assurance Level	EAL1+ (ATE_FUN.1)
Protection Profile	National Database Encryption Protection Profile V1.0
Keywords	Encryption, decryption, DB, database, DBMS, Oracle

1.2. TOE Reference

The TOE that complies with this Security Target is identified as follows:

- Product name: Hancom xDB V2.8
- TOE name: Hancom xDB V2.8
- TOE version: 2.8.2.3
- TOE components and versions

TOE Components	Version	Role	Installation Location
Hancom xDB V2.8 Policy Server	2.8.2.3	Security policy setting, security management setting	Policy Server
Hancom xDB V2.8 APIAgent	2.8.2.3	Perform DB encryption/decryption	Application Server
Hancom xDB V2.8 PluginAgent	2.8.2.3	Perform DB encryption/decryption	Database Server

[Table 1-1] TOE Components and Versions

1.3. TOE Overview

With the development of the Internet, e-business and electronic commerce are actively carried out; information exchange and sharing are becoming more convenient, and security is becoming increasingly important. Until now, however, the main focus has been on the security of communication (session protection) and access control of the system; the stored data, which is the most important object of protection, is stored in plaintext so that no action is taken. Even if you use database system access control to protect your data, the database administrator (DBA) has access to all the data, so there is a limit, i.e., information leak by insiders cannot be prevented. Therefore, there is a need for a solution that can encrypt and protect the stored data.

Hancom xDB V2.8 (”TOE”) encrypts the database ("DB") to prevent unauthorized exposure of the information you want to protect.

The encryption object of the TOE is a DB managed by the database management system (“DBMS”) in the organization’s operating environment. In this Security Target, all data before·and after being encrypted and stored in the DB are defined as user data. Depending on the security policy of the organization operating the TOE, some or all of the user data may be subject to encryption.

The DBMS that manages the DB in the organization’s operating environment is distinct from the DBMS used directly by the TOE to manage the TSF data (security policies, audit data, etc.).

The TOE consists of the Hancom xDB V2.8 Policy Server installed and operated in the policy server, the Hancom xDB V2.8 APIAgent installed and operated in the application server, and the Hancom xDB V2.8 PluginAgent installed and operated in the database server.

1.3.1. Product Use and Main Security Characteristics

The TOE is used by the authorized administrator to encrypt the user data subject to protection to prevent unauthorized exposure of the information to be protected. The authorized administrator can set the algorithm type and key to encrypt and decrypt user data through the encryption policies. User data subject to encryption policies will receive encryption and decryption services provided by the TOE.

The TOE provides the following: (1) security audit function that records and manages the audit data on major auditable events so that the authorized administrator can safely operate the TOE in the organization’s operating environment; (2) encryption key support features such as encryption key management and cryptographic operations for user and TSF data encryption; (3) user data protection to encrypt user data and protect residual information; (4) identification and authentication functions such as authorized administrator identity verification, authentication failure handling, and mutual authentication between TOE components; (5) security management function for security function, role definition, environment settings, etc.; (6) TSF protection functions such as protection of TSF data transmitted between TOE components, protection of TSF data stored in repositories controlled by TSF, and TSF self-tests; and (7) TOE access function for access session management of authorized administrator(s).

The data encryption key (DEK), which is used to encrypt and decrypt user data, is protected by encrypting it with a key encryption key (KEK). The requirements on how to create and use DEK and KEK are described in Section 5.1.2. “Encryption Support (FCS).”

The main functions of each component are as follows:

- Hancom xDB V2.8 Policy Server

The Hancom xDB V2.8 Policy Server ("Policy Server") will be located in the environment that uses cryptographic services. Therefore, it is independent of the environment where data is stored. The Hancom xDB V2.8 APIAgent receives the data encryption and decryption service request from the client used and provides the policy set by the security administrator. The Policy Server is a security management web interface that provides the authorized administrator with security management functions of the TOE. The security administrator sets the policies required by the API through the Policy Server. To this end, the Policy Server stores the main policies, encryption keys, and audit logs in the DBMS connected to the Policy Server. It also records and creates data on the audit logs delivered from Hancom xDB V2.8 APIAgent and Hancom xDB V2.8 PluginAgent.

- Hancom xDB V2.8 APIAgent

  The Hancom xDB V2.8 APIAgent (“APIAgent”) requests encryption and decryption policies from the Policy Server to perform encryption and decryption. APIAgent provides functions of receiving data encryption, decryption, and digest services. It also provides libraries suitable for various developer environments such as C and Java. Even if the encryption and decryption service is requested through APIAgent, the service request is denied if the security administrator has not set the policy or has changed it for security reasons. Moreover, when using encryption and decryption, the audit log is sent to the Policy Server.

- Hancom xDB V2.8PluginAgent

  The Hancom xDB V2.8PluginAgent (“PluginAgent”) is a PluginAgent installed in the database server; it is in charge of receiving the user data delivered from the application server when encryption is in progress, performing DB encryption and decryption according to the policy of the Policy Server at the time of DBMS storage.
  PluginAgent is a combination of APIAgent and reference to the target DBMS.

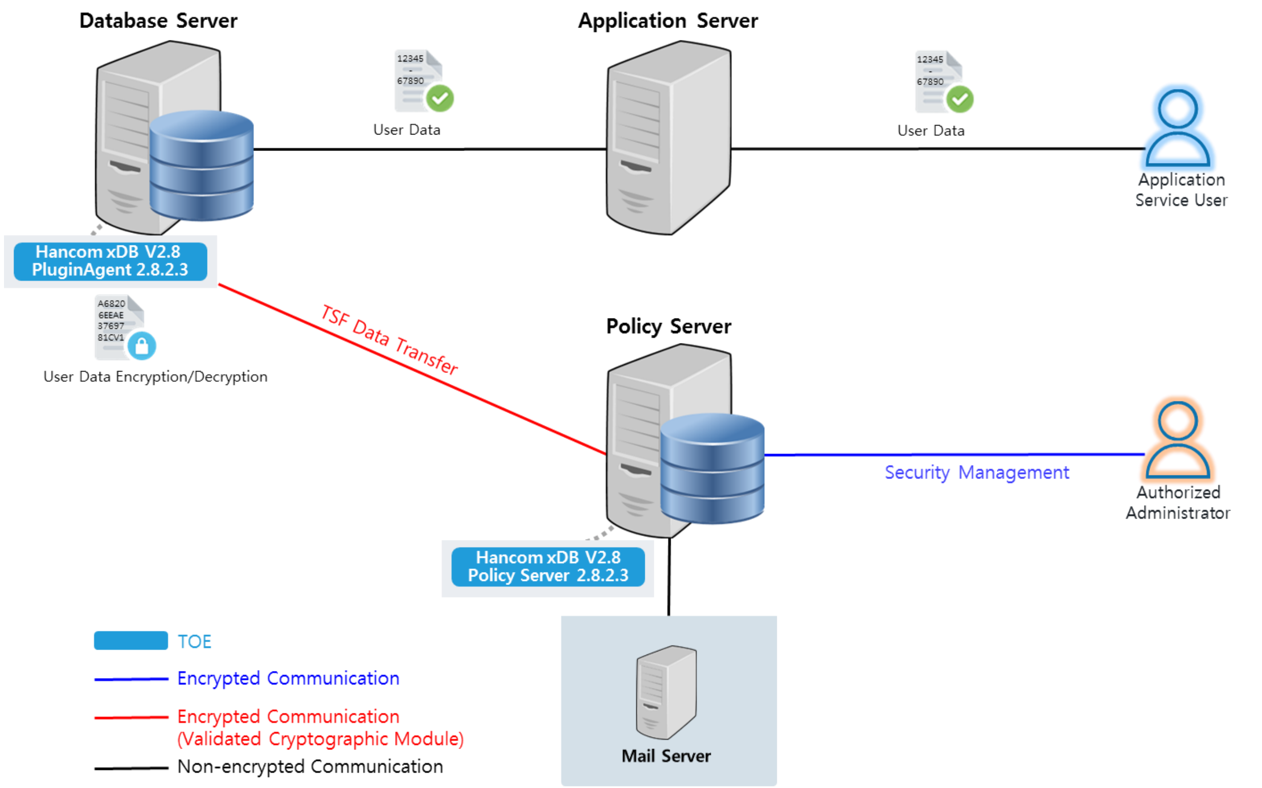
1.3.2. TOE Type

TOE is a “DB encryption” product that encrypts the database (“DB”) to prevent unauthorized exposure of the information to be protected. Each component of the TOE is provided in software format.

The operating environment of the TOE is divided into a “Plugin method” and an “API method” according to the operating method, and both methods are supported. The plug-in method consists of the PluginAgent installed in the database server where the protected DB exists and the Policy Server as the management server. The API method consists of APIAgent and Policy Server as the management server.

The TOE is divided into the main functions such as encryption and decryption function, audit log generation function, audit log and TSF data transmission, and reception function and is implemented as several TOE components.

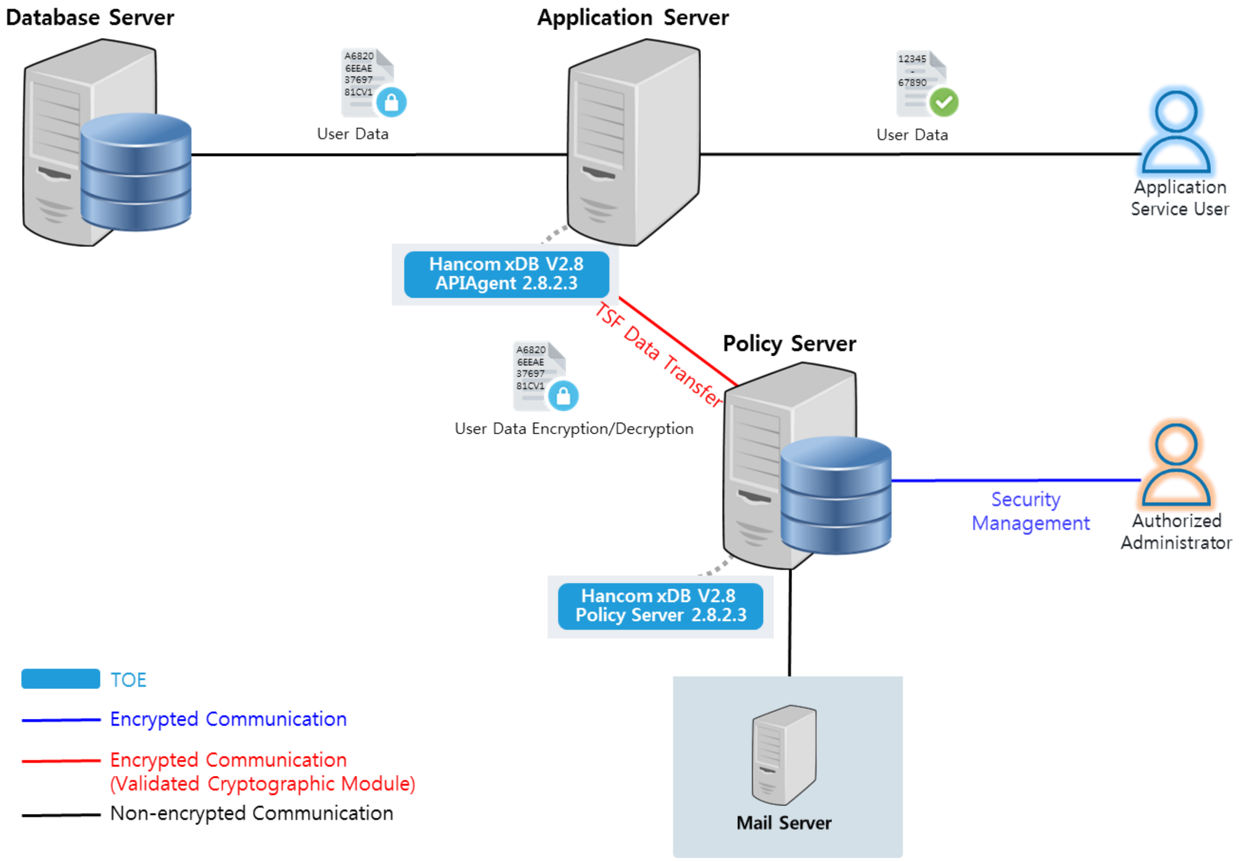
[Figure 1-1] is a block diagram for operating the plug-in method in the TOE's operating environment. It consists of the PluginAgent installed in the database server where the target DB exists and the Policy Server as the management server.



[Figure 1-1] Operating Environment Diagram (Plug-in Method)

PluginAgent encrypts the user data received from the application server according to the authorized administrator's policy before storing it in the DB and decrypts the encrypted user data when the application service user calls the user data through the database server.

[Figure 1-2] is a block diagram of the TOE operating environment when the API is operated. As the TOE component installed in the application server and which provides the application service, APIAgent performs encryption and decryption. It is a library that sends audit logs for encryption and decryption to the Policy Server. It is installed and operated with the Policy Server, which delivers the security management policy for the TOE and collects, records, and manages the audit log.



[Figure 1-2] Operating Environment Diagram (API Method)

The APIAgent module is installed in the application server, performing encryption and decryption of user data according to the authorized administrator's policy. The user data entered by the application service user is encrypted by the APIAgent module installed in the application server and transmitted to the database server. The encrypted user data transmitted from the database server is decrypted by the APIAgent module installed in the application server and transmitted to the application service user.

The authorized administrator can perform encryption and decryption of user data according to the range of encryption objects required by the organization's security policies through the management server. In addition, the authorized administrator can access the management server to perform security management.

External IT entities required to operate the TOE include a mail server for administrator notification functions such as sending audit data loss prediction mails and an NTP server for trusted timestamps. External IT entities and work systems, except the TOE, correspond to the operational environment of the TOE.

1.3.3. Non-TOE Hardware and Software Identification

Additional hardware and software are required to operate the TOE but are not included in the target of evaluation.

The following are the hardware and software requirements for operating the TOE:

(1) Minimum system requirements for TOE operation

TOE	OS	Division	Minimum Requirements
Hancom xDB V2.8 PolicyServer	Linux	OS	CentOS 6.10 kernel 2.6.32 (64 bit)
		CPU	Xeon E3-1220 3.1 GHz or faster
		RAM	16 GB or more
		HDD	Disk space for TOE installation and operation: 100 GB or more
		NIC	1 port or more of 10/100/1000 Ethernet card
Hancom xDB V2.8 APIAgent	Linux	OS	CentOS 6.10 kernel 2.6.32 (64 bit)
		CPU	Xeon E3-1220 3.1 GHz or faster
		RAM	16 GB or more
		HDD	Disk space for TOE installation: 10 MB or more
		NIC	1 port or more of 10/100/1000 Ethernet card
Hancom xDB V2.8 PluginAgent	Linux	OS	CentOS 6.10 kernel 2.6.32 (64 bit)
		CPU	Xeon E3-1220 3.1 GHz or faster
		RAM	16 GB or more
		HDD	10 MB or more space required for TOE installation
		NIC	1 port or more of 10/100/1000 Ethernet card
	AIX	OS	AIX 5.3 64 bit
		CPU	PowerPC POWER64.2 GHz or faster

TOE	OS	Division	Minimum Requirements
		RAM	16 GB or more
		HDD	10 MB or more space required for TOE installation
		NIC	1 port or more of 10/100/1000 Ethernet card
	HP-UX	OS	HP-UX 11.23 64 bit
		CPU	Intel Itanium (IA64)1.4 GHz or faster
		RAM	16 GB or more
		HDD	10 MB or more space required for TOE installation
		NIC	1 port or more of 10/100/1000 Ethernet card

[Table 1-2] Hardware requirements for the operational environment for the TOE

(2) Minimum requirements for the administrator's system for security management

TOE	Division	Minimum Requirements
H/W	CPU	Intel ® Core™ i5-7500 3.4 GHz or faster
	Memory	4 GB or more
	HDD	1000 GB or more
	NIC	1 port or more of 10/100/1000 Ethernet card
S/W	OS	Windows 7 Professional SP1(x86/x64) Windows 10 Pro (x86/x64)
	Web Browser	Chrome 76.0

[Table 1-3] Hardware and software requirements for the administrator's system

(3) Non-TOE software required for TOE operation

TOE	S/W	Purpose
Hancom xDB V2.8 PolicyServer	Java(JDK) 1.8.0_212	Server running and operation based on Java application, security management function, perform web server runs
	Apache Tomcat/7.0.55	Performs encrypted communication between web browsers and servers on the administrator system Web server for providing the server security management screen
	Oracle 11gR1	DBMS for TOE management
Hancom xDB V2.8APIAgent	Java(JDK) 1.8.0_212	Running and operating the TOE’s APIAgent based on Java application

TOE	S/W	Purpose
Hancom xDB V2.8PluginAgent	Java(JDK) 1.6.0 (AIX5.3)	TOE PluginAgent based on Java application, running and operation
	Java(JDK) 1.7.0.27 (HP-UX 11.23)	
	Java(JDK) 1.8.0_212 (CentOS 6.10 kernel 2.6.32)	
	Oracle 11gR1	DBMS for TOE installation

[Table 1-4] Identification and Description of Non-TOE Software Required for TOE Operation

(4) External IT entity used but is not the target of evaluation

Division	TOE Support Functions
Mail Server (SMTPServer)	Server for sending mail to authorized administrators in detecting potential security violations
DBMS	DBMS for storing audit logs of the 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 the TOE consists of the PolicyServer that performs security management such as policy setting, the APIAgent installed in the application server, the PluginAgent installed in the database, and the manuals required for installation and operation. The detailed information is shown in [Table 1-10].

(1) TOE Details

<table>
<tr><th>Division</th><th colspan="2">Identification</th><th>Format</th><th>Distribution</th></tr>
<tr><td>TOE</td><td colspan="2">Hanacom xDB V2.8 (Version: 2.8.2.3)</td><td rowspan="4">S/W</td><td>-</td></tr>
<tr><td rowspan="3">TOE Component</td><td colspan="2">Hancom xDB V2.8 Policy Server 2.8.2.3
(z_package.Hancom_xDB_V2.8_Policy_Server.2.8.2.3.tar.gz)</td><td>CD</td></tr>
<tr><td colspan="2">Hancom xDB V2.8 APIAgent2.8.2.3
(z_package.Hancom_xDB_V2.8_APIAgent.2.8.2.3.Linux.x86_64.64bit.tar.gz)</td><td rowspan="2">CD</td></tr>
<tr><td colspan="2">Hancom xDB V2.8 PluginAgent2.8.2.3
- (z_package.Hancom_xDB_V2.8_PluginAgent.oracle.2.8.2.3.Linux.x86_64.64bit.tar.gz)
- (z_package.Hancom_xDB_V2.8_PluginAgent.oracle.2.8.2.3.AIX.5.3.tar.gz)
- (z_package.Hancom_xDB_V2.8_PluginAgent.oracle.2.8.2.3.HP-UX_IA.B.11.23.tar.gz)</td></tr>
<tr><td rowspan="2">Manual</td><td>Preparative Procedure</td><td>Hancom xDB V2.8 Preparative Procedure (PRE) v1.9
(Hancom xDB_V2.8_Preparative Procedure (PRE) _v1.9.pdf)</td><td rowspan="2">Electronic document (PDF)</td><td rowspan="2">CD</td></tr>
<tr><td>Operation Guide</td><td>Hancom xDB V2.8 Operation Guide (OPE) v1.8
(Hancom xDB V2.8_Operation Guide (OPE) _v1.8.pdf)</td></tr>
</table>

[Table 1-6] Physical Scope of the TOE

(2) Validated Cryptographic Module

General information on the validated cryptographic module used for the TOE is as follows:

TOE	S/W	Purpose
Hancom xDB V2.8 Policy Server	XecureCrypto 2.0.1.1	Validated cryptographic modules for key generation, revocation and renewal, and cryptographic operation, validated cryptographic module for encrypted communication between TOE components
Hancom xDB V2.8APIAgent / Hancom xDB V2.8PluginAgent	XecureCrypto 2.0.1.1	Validated cryptographic modules for key generation, revocation and renewal, and cryptographic operation, validated cryptographic module for encrypted communication between TOE components

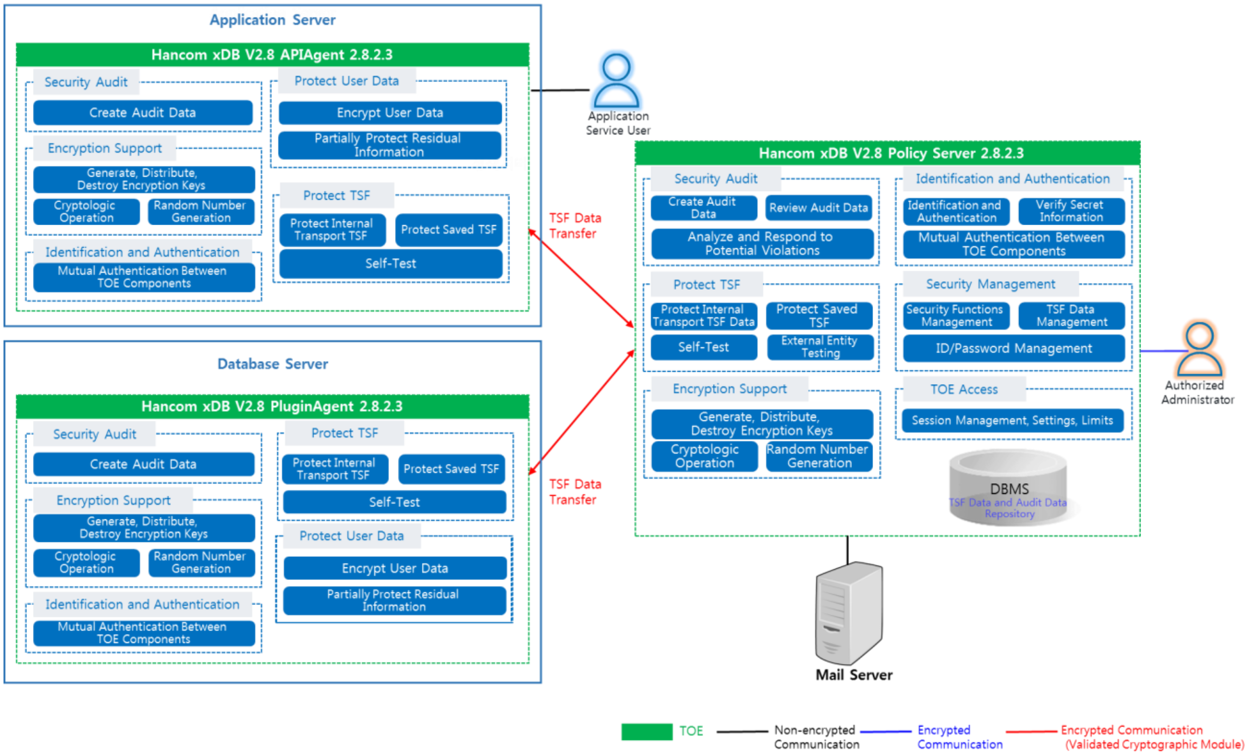
[Table 1-7] Validated Cryptographic Modules in General

The following are detailed information on the validated cryptographic module included in the TOE:

Division	Details
Cryptographic Module Name	XecureCrypto 2.0.1.1
Validation Number	CM-153-2024.5
Validation Level	VSL1
Developer	HANCOM SECURE Inc.
Validation Date	May 2, 2019

[Table 1-8] Validated Cryptographic Modules in General

1.4.2. Logical Scope of the TOE


Application Server
Hancom xDB V2.8 APIAgent 2.8.2.3
Security Audit
Create Audit Data
Encryption Support
Generate, Distribute, Destroy Encryption Keys
Cryptologic Operation
Random Number Generation
Identification and Authentication
Mutual Authentication Between TOE Components
Protect User Data
Encrypt User Data
Partially Protect Residual Information
Protect TSF
Protect Internal Transport TSF
Protect Saved TSF
Self-Test
Application Service User
TSF Data Transfer
Database Server
Hancom xDB V2.8 PluginAgent 2.8.2.3
Security Audit
Create Audit Data
Encryption Support
Generate, Distribute, Destroy Encryption Keys
Cryptologic Operation
Random Number Generation
Identification and Authentication
Mutual Authentication Between TOE Components
Protect TSF
Protect Internal Transport TSF
Protect Saved TSF
Self-Test
Protect User Data
Encrypt User Data
Partially Protect Residual Information
TSF Data Transfer
Hancom xDB V2.8 Policy Server 2.8.2.3
Security Audit
Create Audit Data
Review Audit Data
Analyze and Respond to Potential Violations
Identification and Authentication
Identification and Authentication
Verify Secret Information
Mutual Authentication Between TOE Components
Protect TSF
Protect Internal Transport TSF Data
Protect Saved TSF
Self-Test
External Entity Testing
Security Management
Security Functions Management
TSF Data Management
ID/Password Management
Encryption Support
Generate, Distribute, Destroy Encryption Keys
Cryptologic Operation
Random Number Generation
TOE Access
Session Management, Settings, Limits
DBMS
TSF Data and Audit Data Repository
Authorized Administrator
Mail Server
🟩 TOE
Non-encrypted Communication
Encrypted Communication
Encrypted Communication (Validated Cryptographic Module)

[Figure 1-3] Logical Scope of the TOE

1) Security Audit

The TOE creates and maintains audit records of auditable events such as the operation of 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 the security management server of the TOE, the Policy Server generates audit data on audited events that occur from the management function (including security management) and stores them in the DBMS. In addition, audit data is generated for the audit target events including encryption and decryption of user data performed by APIAgent and PluginAgent and is transmitted to the Policy Server. The Policy Server stores the DBMS audit data received from APIAgent and PluginAgent.

Upon detecting a potential security breach, the TOE takes actions in response to the security breach. It emails all authorized administrators of all potential violation audits.

The TOE provides the function of selectively reviewing the audit data generated by the TOE and stored in the DBMS. It selectively reviews the saved API audit log according to the inquiry date, encryption policy ID, encryption account, operation type, and results. The administrator audit log is selectively reviewed according to the task type, task operation, worker, results, inquiry period, and CLIENT IP. Authorized administrators can selectively review each audit log.

All the audit data generated are stored in the audit trace storage (DBMS) for safe management; unauthorized deletion of audit data is prevented, with a function of protecting audit trace storage by ignoring the audited events when the audit trace is saturated.

2) Cryptographic Support

The TOE provides cryptographic key generation, distribution, revocation, and cryptographic operations to protect the transmitted data between TOE components and encrypt and decrypt user data. In addition, it provides a random number generation function for secure encryption key generation. The TOE uses the encryption target cryptographic algorithm of "XecureCrypto v2.0.1.1," a validated cryptographic module whose security and implementation conformance are verified through the cryptographic module verification system (KCMVP) for the encryption of user data and TSF data to generate the encryption key. If no longer needed, the encryption key is deleted and overwritten by zeroing it to destroy it.

The TOE performs a cryptographic operation using the verification target cryptographic algorithm of "XecureCrypto v2.0.1.1," a validated cryptographic module whose security and implementation conformance are verified through the cryptographic module verification system (KCMVP) for user data encryption. When performing the operation, the validated cryptographic module operates operates 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 plaintext; the counter used in IV, CTR mode and CBC, CFB, and OFB modes applies the method described in KS X 1213 and TTAS.KO-12.004/R1. The TOE performs cryptographic operation using the verification target cryptographic algorithm of "XecureCrypto v2.0.1.1," a validated cryptographic module whose security and implementation conformance are verified through the cryptographic module verification system (KCMVP) for TSF data encryption. When performing the cryptographic operation, the validated cryptographic module operates 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 plaintext; the counter used in the IV and CBC modes applies the method described in KS X 1213 and TTAS.KO-12.004/R1.

When using a random number in the SFR that requires the use of the verification target cryptographic algorithm of the cryptographic module, such as generating the main cryptographic key including a key for user data encryption (DEK), the TOE uses the random number generator of “XecureCrypto v2.0.1.1,” a validated cryptographic module whose safety and implementation conformity are verified through the cryptographic module verification system (KCMVP).

- Cryptographic key generation:
  - HASH_DRBG (SHA256, 256 bit): 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)
  - RSAES (2048 bit): Asymmetric key generation for self-implemented communication
- Encryption key distribution
  - Symmetric Key Encryption and Decryption (ARIA-CBC, 128 bit): Performs data encryption and decryption between TOE components in self-implemented communications
  - ARIA-CBC (128 bit) + RSAES (2048 bit): Uses TSF data protection for mutual authentication
- Cryptographic operations
  - Symmetric Key Encryption and Decryption (ARIA-CBC, 128 bit): TSF data, encryption key (KEK, DEK) encryption and decryption
  - Symmetric Key Encryption and Decryption (ARIA-CBC, 128 bit / 192 bit / 256 bit): User Data Encryption and Decryption
  - One-Way Encryption (SHA256): TSF data encryption, integrity validation
  - One-Way Encryption (SHA224, 256, 384, 512): user data encryption
- Cryptographic Key Destruction
  - After sending the encryption key from Hancom xDB V2.8 Policy Server to Hancom xDB V2.8 APIAgent and Hancom xDB V2.8 PluginAgent, update the temporarily stored encryption key information to 0x00 and destroy the encryption key information.

3) User Data Protection

The TOE provides the function of encrypting and decrypting user data by column.

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

When performing user data encryption and decryption, key and policy information is deleted from memory after use and requested every time.

4) Identification & Authentication

To allow access to the security management functions provided by the TOE, the authorized administrator must be successfully identified and authenticated before allowing all actions related to the security functions.

The identity of the authorized administrator is verified based on ID and password. The password of the security administrator must be set to 9 ~ 20 characters including all 4 uppercase letters, lowercase letters, numbers, and special characters (!,@, #,%, ^,*, (,), -, =, , +, [,], {, }; ,., /, >, ?). .

If 5 consecutive authentications fail during identification and authentication for the authorized administrator's administrative access, it blocks administrative access to the account for 5 minutes and stores an audit record of the authentication failure.

To protect authentication feedback, the TOE masks the password that is inputted when logging in to the TOE security management, adding administrators, and modifying information. In addition, when identification and authentication fail, it does not provide feedback on the reason for the failure.

The TOE guarantees the uniqueness of a session ID by using a validated cryptographic module to prevent reuse of authentication data.

Mutual authentication between APIAgent, PluginAgent, and Policy Server is performed through self-implemented authentication protocol. In mutual authentication, the APIAgent and the Policy Server issue a private key and a public key, respectively, and then generate a signature (api-signMessage) with a specific message (api-originMessage) and an API private key (api-privateKey). The Policy Server authenticates the signature (api-signMessage) with a specific message (api-originMessage) and an API public key (api-publicKey). In the same way, the Policy Server generates a signature (pol-signMessage) with a specific message (pol-originMessage) and a Policy Server private key (pol-privateKey). In APIAgent, mutual authentication is performed by authenticating the signature (pol-signMessage) with a specific message (pol-originMessage) and the Policy Server's public key (po; -publicKey).

5) Security Management

The TOE provides the security management function for the authorized administrator to set up and manage the security policy and important data. The security manager must execute the security management function through the security management interface Policy Server and can use the security management function only after the identification and authentication process. Security management functions can be divided into administrator account management, key management, policy management, and security management interface settings.

6) Protection of the TSF

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.2.0.1.1," 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.

The TOE protects the passwords, encryption keys, critical security parameters, TOE configuration values (security policies, configuration parameters), and audit data of authorized administrators and DB encryption users stored in the TSF data repository from unauthorized exposures and modifications. In particular, TSF data such as passwords of authorized administrators and DB encryption users, data encryption keys (DEK), critical security parameters, TOE configuration values, DBMS connection information, etc. are encrypted and stored with the verification object encryption algorithm of the verified encryption module "XecureCrypto v.2.0.1.1.".

The data encryption key (DEK) should be encrypted and stored using the encryption target encryption algorithm provided by the encrypted module using the key encryption key (KEK). KEK is encrypted and saved using the encryption file.

Encryption keys and critical security parameters loaded into the memory do not exist as plain text in memory when the encryption and decryption operation is complete and not used.

The TOE executes self-tests periodically at startup and during normal operation to verify the correct operation of the Policy Server, PluginAgent, and APIAgent.

The core process of performing the TSF is subject to self-tests; the validated cryptographic modules can also receive self-test results and point out potential violations in their own tests.

The TOE provides the function of verifying the integrity of TSF data and TSF such as main executable files, configuration files, etc. Integrity checks are performed at startup, periodically during normal operation, and as desired by the authorized administrator.

In case of violation from the integrity check results, the authorized administrator is notified by e-mail.

Hancom xDB V2.8 Policy Server performs integrity validations on all files below ROOT in the path where the Policy Server is installed. Note, however, that log files are excluded from integrity validations.

The following is a list of integrity files checked by Hancom xDB V2.8 APIAgent:

Type	Name	Description
Config file	xdf.ini	APIAgent environment configuration file
	config.json	APIAgent encryption environment configuration file
Library file	xecuredbapi.jar	Encryption/decryption library (Java)
	libxecuredbapi.so	Encryption/decryption library (C/C++)
	XDBAgent.jar	APIAgent core library
	libXecureASN.so	Module for ASN.1 operations
	libXecureCSP.so	Module that provides an interface between cryptographic libraries and other modules
	libXecureCodec.so	Module that provides a string conversion function
	libXecureIO.so	Module that provides functions related to memory, file, socket, time, etc.
	libXecurePKCS5.so	Password-based encryption
	libXecurePKCS8.so	Module that controls a user's secret key information
	libXecureSSL.so	Module that provides the functions required for SSL or TCPIP communication

Type	Name	Description
	libXecureCrypto.so	Validated cryptographic module interface library
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	APIAgent TSF data encryption key

The list of integrity items to be monitored by Hancom xDB V2.8 PluginAgent is as follows:

Type	Name	Description
Config file	xdf.ini	PluginAgent environment configuration file
	config.json	PluginAgent encryption environment configuration file
Library file	xecuredbapi.jar	Encryption/decryption library (Java)
	libxecuredbapi.so	Encryption/decryption library (C/C++)
	XDBAgent.jar	PluginAgent core library
	libxdbplugin_comm_jni.so	Oracle Plugin library
	xdbplugin_comm_jni.class	Class for running Hancom xDB V2.8 API (Java) in Oracle
	libXecureASN.so libXecureASN.sl	Module for ASN.1 operation (OS dependable extensions)
	libXecureCSP.so libXecureCSP.sl	Module that provides an interface between cryptographic libraries and other modules (OS dependable extensions)
	libXecureCodec.so libXecureCodec.sl	Module that provides a string conversion function (OS dependable extensions)
	libXecureIO.so	Module that provides functions related to memory, file, socket, time, etc.

Type	Name	Description
	libXecurePKCS5.so libXecurePKCS5.sl	Password-based encryption (OS dependable extensions)
	libXecurePKCS8.so libXecurePKCS8.sl	Module that controls a user's secret key information (OS dependable extensions)
	libXecureSSL.so libXecureSSL.so	Module that provides the functions required for SSL or TCPIP communication (OS dependable extensions)
	libXecureCrypto.so libXecureCrypto.sl	Validated cryptographic module interface library (OS dependable extensions)
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	PluginAgent TSF data encryption key

Hancom xDB V2.8 Policy Server checks the normality of the external entities DBMS and mail server, which are necessary to operate the TOE, when it is running. The DBMS judges whether it is normal by using the SQL query, and the mail server judges whether it is normal by sending the test mail. If the external entities are not normal, the TOE is terminated to disable the service.

7) TOE access

The TOE blocks the maximum number of concurrent sessions to 1 to disable concurrent login from the same account. The TOE also blocks simultaneous access to the same authorization. If you try to access the same account or the same authorization at the same time, the new connection is blocked, and the existing connection is maintained.

The TOE terminates the session if there is no activity for 5 minutes after the authorized administrator logs in.

The TOE controls access to it so that only the registered IP (two default values or less) can access the security management interface. After the TOE installation, access-permitted IPs can be set when logging in for the first time; after installation, you can add, change, or delete the access-permitted IPs through the security management login-allowed IP list setting. When setting the IP, you cannot add the IP address range; you must add one IP address individually. At this time, setting 0.0.0.0, 192.168.10.*, any, etc., which means the entire network range, is not allowed.

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 <u>*underlined italics*</u>.

- **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 declaration of the Common Criteria, Protection Profile, or package conformed to by this Security Target as well as how the Protection Profile or Security Target conforms to the Protection Profile.

2.1. Common Criteria Conformance

This Security Target expands from Part 2 of the V3.1 Amendment Volume 5 and complies with Part 3 of the V3.1 Amendment Volume 5 to the Computer Security System Common Criteria (Notice of the Ministry of Science, ICT, and Future Planning, No. 2013-51).

Common Criteria		Common Criteria for Information Security System version 3.1 revision 5 - Common Criteria for Information Security System, Part 1: Introduction and general model, Version 3.1r5 (CCMB-2017-04-001, April 2017) - Common Criteria for Information Security System, Part 2: Security functional components, Version 3.1r5 (CCMB-2017-04-002, April 2017.) - Computer Security System Common Criteria, Part 3: Security assurance components, Version 3.1r5 (CCMB-2017-04-003, April 2017)
Protection Profile		National Database Encryption Protection Profile V1.0
Compliance Form	Part 2 Security functional requirements	Extension: FCS_RBG.1, FIA_IMA.1, FDP_UDE.1, FMT_PWD.1, FPT_PST.1, FTA_SSL.5
	Part 3 Security assurance requirements	In compliance
	Package	Add: Add EAL1 (ATE_FUN.1)

2.2. Protection Profile Compliance

This Security Target complies with the same security objectives and security requirements for the operating environment through strict compliance with the National Database Encryption Protection Profile V1.0 (KECS-PP-0820-2017).

2.3. Package Conformance

The assurance requirements package conformed to by this Security Target is EAL1, and some additional assurance requirements are additionally defined.

- Assurance package: Added EAL1 (ATE_FUN.1)

2.4. Rationale of Declaration of Compliance

This Security Target conforms to the TOE type, security objectives, and security requirements of the Protection Profile in the same way. Thus, the declaration of compliance for "National Integrated Certification Protection Profile V1.0" is "Strict Protection Profile Compliance."

Rationale of security objectives added in accordance with the "National Database Cryptography Protection Profile V1.0" selection SFR

Item	Security Objectives	Rationale
Security Objectives for the Operational Environment	OE. Time stamp	Security objective added for the selected SFR FPT_STM.1
	OE. DBMS	Security objective added for the selected SFR FAU_STG.1
	OE. Trusted path/channels	Security objective added for the SFR FTP_TRP.1

3. Security Objectives

This Security Target defines the security objectives by classifying them into TOE security objectives and security objectives for the operational environment. The security objectives for the TOE are those handled directly by the TOE, and the security objectives for the operational environment are those handled by the IT environment or by non-technical / procedural means.

3.1 Security Objectives for the Operational Environment

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

OE. Physical Security

The place where the TOE is installed and operated should have access control and protection facilities accessible only to authorized administrators.

OE. Trusted Administrator

The authorized administrator of the TOE is not malicious and must be properly trained on the TOE management functions, performing the duties correctly according to the administrator guidelines.

OE. Safe Development

Developers interworking encryption functions into applications or DBMS using the TOE should ensure that the security functions of the TOE are applied securely by complying with the requirements of the guidance document provided together with the TOE.

OE. Log Backup

The authorized administrator of the TOE should periodically check the free space of the audit data storage in preparation for loss of audit records and back up the audit records (external log server, separate storage device, etc.) to prevent the audit records from being deleted.

OE. Operating System Enhancement

The authorized administrator of the TOE should ensure the reliability and safety of the operating system by addressing the latest vulnerabilities of the operating system on which the TOE is installed and operated.

OE. Time stamp

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

OE. DBMS

In order to protect the repository where TSF data is stored, the DBMS should be installed and operated to block all connections except the TOE.

OE. Trusted path/channels

The TOE should protect all information transmitted when the authorized administrator accesses the management server through a web browser via a secure path / channel.

4. Extended Components Definition

In addition to the components in Part 2 of the Common Criteria, this Security Target additionally defines the components below. The extended components of this Security Target are as follows:

- Cryptographic Support (FCS)
  - FCS_RBG.1 Random bit generation

- Identification & Authentication (FIA)
  - FIA_IMA.1 TOE internal mutual authentication

- User Data Protection (FDP)
  - FDP_UDE.1 User data encryption

- FMT, Security Management
  - FMT_PWD.1 Management of ID and password

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

- TOE Access (FTA)
  - FTA_SSL.5 Management of TSF-initiated sessions

4.1. Cryptographic Support (FCS)

4.1.1. Random Bit Generation

1) Family Overview

The random number generation (FCS_RBG, Random Bit Generation) family is defined to require the function of generating the random values required for the TOE cryptographic operation.

2) Hierarchy and Description of the Component(s)

FCS_RBG Random bit generation
1

FCS_RBG.1 Random number generation requires the function of the TSF to generate random values for the TOE cryptographic operations.

* Management: FCS_RBG.1
No expected management requirements

* Audit: FCS_RBG.1
There are no auditable events foreseen.

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

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

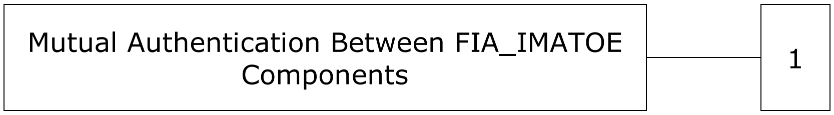
4.2. Identification & Authentication (FIA)

4.2.1. TOE internal mutual authentication

1) 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.

2) Hierarchy and Description of the Component(s)



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
If the FAU_GEN security audit data generation family is included in the Protection Profile / Security Target, auditing the following actions is recommended:

a) Minimum: Success and failure of mutual authentication

b) Minimum: Change in the authentication protocol

FIA_IMA.1 TOE internal mutual authentication
Hierarchical to: No other components
Dependencies: No dependencies

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

4.3. User Data Protection (FDP)

4.3.1. User Data Encryption

1) Family Behaviour

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

2) Component levelling

FDP_UDE User data encryption

1

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

*Management: FIA_UDE.1
The following actions could be considered for the management functions in FMT:
a) Management of user data encryption/decryption rules

* Audit: FDP_UDE.1
The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: User data encryption/decryption failure

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 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 and Password

1) Family Behaviour

The family defines the requirements for functions that should be available to authorised users with regard to the management of the ID and password used in the TOE and their setting or modification.

2) Component levelling

FMT_PWD ID and Password

1

FMT_PWD.1 The management of ID and Password requires the TSF to provide functions for managing the ID and password.

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

* Audit: FMT_PWD.1
The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:
a) Minimal: All modifications to password

FMT_PWD.1 Management of ID and Password
Hierarchical to : No other components
Dependencies : FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

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

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

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

1. [assignment: *combination rules and/or length of the ID*]

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

FMT_PWD.1.3 The TSF shall provide the capability regarding [selection, choose one of: *set the ID and password during installation, set password during installation, change the ID and password upon the authorized administrator's first login, change the password upon the authorized administrator's first login*].

4.5. Protection of the TSF (FPT)

4.5.1. Basic Protection of Stored TSF Data

1) Family Behaviour

This family defines the rules for protecting the TSF data stored within the containers controlled by the TSF from unauthorized modification or disclosure.

2) Component levelling

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

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

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

FPT_PST.1 Basic Protection of Stored TSF Data
Hierarchical to: No other components
Dependencies: No dependencies

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

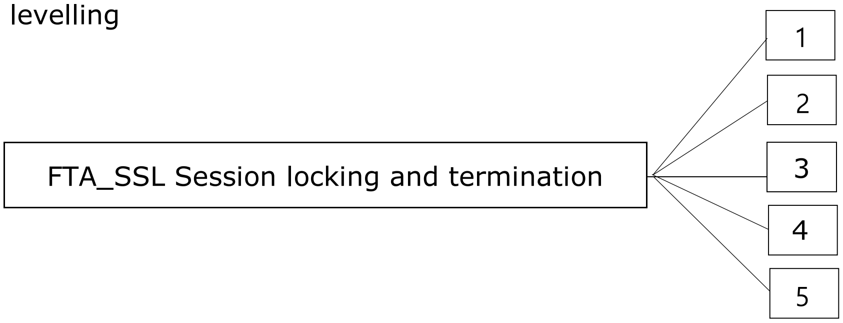
4.6. TOE Access (FTA)

4.6.1. Session Locking and Termination

1) Family Behaviour

This family defines the requirements for the TSF to provide the capability for TSF-initiated and user-initiated locking, unlocking, and termination of interactive sessions.

2) Component levelling



In CC Part 2, the Session locking and termination family consists of four components. In this PP, it contains five components with one additional extended component as follows:
※ The relevant descriptions of four components contained in CC Part 2 are omitted.

FTA_SSL.5 TSF-initiated termination provides the requirements for the TSF to lock or terminate the session after a specified period of user inactivity.

* Management: FTA_SSL.5
The following actions could be considered for the management functions in FMT:

a) Specification of the time of user inactivity after which lock-out or termination occurs for an individual user;
b) Specification of the default time of user inactivity after which lock-out or termination occurs

* Audit: FTA_SSL.5
The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:
a) Minimal: Locking or termination of an interactive session

FTA_SSL.5 Management of TSF-initiated sessions
Hierarchical to: No other components
Dependencies: No dependencies

FTA_SSL.5.1 The TSF shall carry out [selection: *lock an interactive session and/or re-authenticate the user before unlocking the session,* *terminate the session*] after [assignment: time interval of user inactivity].

5. Security Requirements

This chapter describes the security functional requirements and assurance requirements that should be met by the TOE.

5.1. Security Functional Requirements

The security functional requirements defined in this Security Target were selected and used from Part 2 of the Common Criteria and the extended component definitions of Chapter 4.

Security Function Class	Security Function Component	
Security Audit (FAU)	FAU_ARP.1	Security alarm
	FAU_GEN.1	Audit data generation
	FAU_SAA.1	Potential violation analysis
	FAU_SAR.1	Audit review
	FAU_SAR.3(1)	Selectable audit review (API audit log)
	FAU_SAR.3(2)	Selectable audit review (Administrator audit log)
	FAU_STG.3	Action in case of possible audit data loss
	FAU_STG.4	Prevention of audit data loss
Cryptographic Support (FCS)	FCS_CKM.1(1)	Cryptographic key generation (User data encryption)
	FCS_CKM.1(2)	Cryptographic key generation (Encryption for TSF data/Mutual authentication)
	FCS_CKM.2	Cryptographic key distribution
	FCS_CKM.4	Cryptographic key destruction
	FCS_COP.1(1)	Cryptographic operation (User data encryption)
	FCS_COP.1(2)	Cryptographic operation (TSF data encryption/Mutual authentication)
	FCS_RBG.1(Extended)	Random bit generation
User data protection (FDP)	FDP_UDE.1(Extended)	User data encryption
	FDP_RIP.1	Subset residual information protection
Identification & Authentication (FIA)	FIA_AFL.1	Authentication failure handling
	FIA_IMA.1(Extended)	TOE internal mutual authentication
	FIA_SOS.1	Verification of secrets
	FIA_UAU.2	User authentication before any action
	FIA_UAU.4	Single-use authentication mechanisms

Security Function Class	Security Function Component	
	FIA_UAU.7	Protected authentication feedback
	FIA_UID.2	User identification before any action
Security Management (FMT)	FMT_MOF.1	Security function management
	FMT_MTD.1	Management of TSF data
	FMT_PWD.1(Extended)	Management of ID and password
	FMT_SMF.1	Management function specification
	FMT_SMR.1	Security roles
Protection of the TSF (FPT)	FPT_ITT.1	Basic internal TSF data transfer protection
	FPT_PST.1(Extended)	Basic protection of stored TSF data
	FPT_TEE.1	External entity test
	FPT_TST.1	TSF testing
TOE access (FTA)	FTA_MCS.2	Per user attribute limitation on multiple concurrent sessions
	FTA_SSL.5(Extended)	Management of TSF-initiated sessions
	FTA_TSE.1	TOE session establishment

[Table 5-1] Security Functional 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 Audit data generation

Hierarchical to: No other components

Dependencies: FPT_STM.1 Reliable time stamps

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

a) Start-up and shutdown of the audit functions

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

c) [“Auditable Event” of [Table 5-2], *none*]

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

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

b) For each type of audit event, [“Additional Audit Record Contents” in [Table 5-2], *none*] based on the auditable event definition of the functional component are included in the Security Target.

Component ID	Auditable Event	Additional Audit Records
FAU_ARP.1	Response actions taken due to potential security breaches	
FAU_SAA.1	Start and stop of analysis mechanism operation, automatic response by tool	
FAU_STG.3	Response actions when the threshold is exceeded	
FAU_STG.4	Response actions when audit storage fails	
FCS_CKM.1(1)	Success and failure of actions	
FCS_CKM.1(2)	Success and failure of actions	
FCS_CKM.2	Success and failure of actions (Applies only to key distribution related to TSF data encryption and decryption)	
FCS_CKM.4	Success and failure of actions (applies only to key destruction related to TSF data encryption and decryption)	
FCS_COP.1(1)	Success and failure of cryptographic operation, types of cryptographic operation	
FCS_COP.1(2)	Success and failure of cryptographic operation, types of cryptographic operation	
FDP_UDE.1	Success and failure in user data encryption/decryption	
FIA_AFL.1	Reach the threshold of failed authentication attempts and the corresponding actions taken; if appropriate, subsequent recovery to normal status	
FIA_IMA.1	Success or failure of mutual authentication	
FIA_UAU.2	All uses of authentication mechanism	

Component ID	Auditable Event	Additional Audit Records
FIA_UAU.4	Attempts to reuse authentication data	
FIA_UID.1	All uses of administrator identification mechanisms, including provided administrator identity	
FMT_MOF.1	All modifications to the functions in the TSF	
FMT_MTD.1	All modifications to the values of the TSF data	Changed TSF data values
FMT_PWD.1	All changes to the password	
FMT_SMF.1	Use management features	
FMT_SMR.1	Modifications to user groups sharing roles	
FPT_TST.1	Execution of the TSF self-tests and results of the tests	TSF data or execution code changed in case of integrity violation
FPT_TEE.1	Execution of external entity test and test results	
FTA_MCS.2	Denial of a new session based on the restrictions on the number of concurrent sessions	
FTA_SSL.5	Locking or termination of an interactive session	
FTA_TSE.1	Denial of session establishment based on the session establishment mechanism Any attempt to establish a user session	

[Table 5-2] Auditable Cases

FAU_SAA.1 Potential Violation Analysis

Hierarchical to: No other components

Dependencies: FAU_GEN.1 Audit data generation

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

FAU_SAA.1.2 The TSF shall enforce the following rules for monitoring audited events:

a) accumulation or combination of [

authentication failure audit events among auditable events in FIA_UAU.1,

integrity violation audit event among FPT_TST.1 auditable subjects, self-test failure of the validated cryptographic module,
actions in case of possible audit data loss specified in FAU_STG.3,
more than 5 consecutive login failures specified in FIA_AFL.1,
simultaneous attempt to connect to administrator of the same account specified in FTA_MCS.2,
attempt to access security management from an unregistered IP specified in FTA_TSE.1, and simultaneous attempt to connect to administrator with the same authorization]

b) [None]

FAU_SAR.1 Audit review

Hierarchical to : No other components
Dependencies : FAU_GEN.1 Audit data generation

FAU_SAR.1.1 The TSF shall provide [authorized administrators] with the capability to read [all audit data] from the audit records.

FAU_SAR.1.2 The TSF shall provide the audit records in a manner suitable for the authorized administrator to interpret the information.

FAU_SAR.3 (1) Selectable Audit Review (API 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 encryption policy ID, encryption account, task type, and outcome] of audit data based on [AND].

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)	128-bit
				192-bit
				256-bit

[Table 5-3] Cryptographic 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 data encryption key" (Master key)	ISO/IEC 18031	HASH_DRBG(SHA256)	128bit
	Cryptographic key for the "Master key"	ISO/IEC 18031	HASH_DRBG(SHA256)	128bit
	TSF data (Environment configuration file) Cryptographic key	ISO/IEC 18031	HASH_DRBG(SHA256)	128bit
	Cryptographic key for data transfer (Session key)	ISO/IEC 18031	HASH_DRBG(SHA256)	128bit
Mutual authentication	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.2 Cryptographic key distribution

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

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method [RSAES-OAEP] that meets the following: [ISO/IEC 18033-2].

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	APIAgent/PluginAgent : When encryption/decryption is finished Policy Server : When the "Discard" button in the "Cryptographic Key Management" menu is run	APIAgent/PluginAgent : Cryptographic keys are destroyed by overwriting the cryptographic key data with 0x00. Policy Server : Deleted from the cryptographic key storage
Cryptographic key for the "User data encryption key" (Master key)	Policy Server : When the "Discard" button in the "Cryptographic Key Management" menu is run	Deleted from the cryptographic key storage
Cryptographic key for data transfer (Session key):	When encryption/decryption for data transfer is finished	Cryptographic keys are destroyed by overwriting the cryptographic key data with 0x00.

[Table 5-5] Cryptographic key destruction

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]].

Cryptographic Operation	Standards	Cryptographic Algorithm	Cryptographic Key Size
Block Cipher	KS X 1213	ARIA128	128bit
		ARIA192	192bit
		ARIA256	256bit
	TTAS.KO-12.004/R1	SEED	128bit
Hash	ISO/IEC_10118-3	SHA224	224bit
		SHA256	256 bit
		SHA384	384 bit
		SHA512	512 bit

[Table 5-6] Cryptographic algorithm (User data encryption)

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 The TSF shall perform [encryption and decryption for TSF data/mutual authentication] in accordance with a specified cryptographic algorithm [algorithms described in [Table 5-7]] and the cryptographic key sizes [key sizes described in [Table 5-7] that meet the following: [standards described in [Table 5-7]].

Classification	Cryptographic Operation	Standards	Cryptographic Algorithm	Cryptographic Key Size
TSF data encryption	Block Cipher	KS X 1213	ARIA128	128bit
	Hash	ISO/IEC_10118-3	SHA256	256 bit
Mutual authentication	Block Cipher	KS X 1213	ARIA128	128bit
	Hash	ISO/IEC_10118-3	SHA256	256 bit

[Table 5-7] Cryptographic algorithm

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 Verification of secrets

Hierarchical to : No other components
Dependencies : No dependencies

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.

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 Single-use 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].

Security Function	Management Behavior
Policy Management - Policy	Query
	New
	Modify
	Delete

[Table 5-8] List of security management functions

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].

TSF Data List	Management
Administrator's password	Modify
API/Plugin user data	Query
	New
	Modify
	Delete
Allowed IPs for administrator logins	Query
	New
	Modify
	Delete
Cryptographic key group info	Query
	New
	Modify
	Delete
Cryptographic key info	Query
	New
	Modify
	Delete
Encryption rule info	Query
	New

TSF Data List	Management
	Modify
	Delete
Encryption policy info	Query
	New
	Modify
	Delete
API/Plugin Installation IP	Query
	New
	Delete
Cryptographic key for the "Master Key"	Create
Cryptographic key for the "Environment configuration file"	Create
Key pair for mutual authentication	Create

[Table 5-9] 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*.

Division	Type	Name	Description
APIAgent	Library file	xecuredbapi.jar	Encryption/decryption library (Java)
		libxecuredbapi.so	Encryption/decryption library (C/C++)
		XDBAgent.jar	APIAgent core library
		libXecureASN.so	Module for ASN.1 operations
		libXecureCSP.so	Module that provides an interface between cryptographic libraries and other modules
		libXecureCodec.so	Module that provides a string conversion function
		libXecureIO.so	Module that provides functions related to memory, file, socket, time, etc.

Division	Type	Name	Description
		libXecurePKCS5.so	Password-based encryption
		libXecurePKCS8.so	Module that controls a user's secret key information
		libXecureSSL.so	Module that provides the functions required for SSL or TCPIP communication
		libXecureCrypto.so	Validated cryptographic module interface library
	Config file	xdf.ini	APIAgent environment configuration file
		config.json	APIAgent encryption environment 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	APIAgent TSF data cryptographic key
PluginAgent	Library file	xecuredbapi.jar	Encryption/decryption library (Java)
		libxecuredbapi.so	Encryption/decryption library (C/C++)
		XDBAgent.jar	PluginAgent core library
		libxdbplugin_comm_jni.so	Oracle Plugin library
		xdbplugin_comm_jni.class	Class for running Hancom xDB V2.8 API (Java) in Oracle
		libXecureASN.so libXecureASN.sl	Module for ASN.1 operations (OS dependable extensions)
		libXecureCSP.so libXecureCSP.sl	Module that provides an interface between cryptographic libraries and other modules (OS dependable extensions)
		libXecureCodec.so libXecureCodec.sl	Module that provides a string conversion function (OS dependable extensions)
		libXecureIO.so libXecureIO.sl	Module that provides functions related to memory, file, socket, time, etc. (OS dependable extensions)
		libXecurePKCS5.so	Password-based encryption

Division	Type	Name	Description
		libXecurePKCS5.sl	(OS dependable extensions)
		libXecurePKCS8.so libXecurePKCS8.sl	Module that controls a user's secret key information (OS dependable extensions)
		libXecureSSL.so libXecureSSL.so	Module that provides the functions required for SSL or TCPIP communication (OS dependable extensions)
		libXecureCrypto.so libXecureCrypto.sl	Validated cryptographic module interface library (OS dependable extensions)
	Config file	xdf.ini	APIAgent environment configuration file
		config.json	APIAgent encryption environment 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	APIAgent TSF data cryptographic key
Policy Server	The policy server shall perform integrity verification on all files under Hancom_xDB_V2.8/ in the installation path of the policy server. (Excluding Hancom_xDB_V2.8/logs)		

[Table 5-10] TSF data

FPT_TEE.1 Testing of external entities

Hierarchical to: No other components

Dependencies: No dependencies

FPT_TEE.1.1 The TSF shall run a suite of tests during *initial start-up* to check the fulfillment of [

a. Whether a DB query of DBMS was successfully executed

b. Whether a test mail of mail server was successfully sent

].

FPT_TST.1.2 If the test fails, the TSF shall perform [shut-down].

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 restrict the maximum number of concurrent sessions belonging to the same **administrator** according to the rules for the list of management functions defined in FMT_SMF.1.1 as follows:

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

b) limit the maximum number of concurrent sessions to {0, since there is no "monitoring only" administrator privilege} for administrative access by the same administrator who does not have the right to perform FMT_MOF.1.1 Management behaviour but has the right to perform only the query in FMT_MTD.1.1 Management

c) [none]

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

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

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

FTA_SSL.5.1 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*].

5.2. 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 Evaluation	ASE_INT.1	ST introduction
	ASE_CCL.1	Conformance claims
	ASE_OBJ.1	Security objectives for the operational environment
	ASE_ECD.1	Extended components definition
	ASE_REQ.1	Stated security requirements
	ASE_TSS.1	TOE summary specification
Development	ADV_FSP.1	Basic functional specification
Guidance documents	AGD_OPE.1	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 5-10] 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 table below shows the dependencies of security functional requirements.

No.	Security Function Component	Dependencies	Reference No.
1	FAU_ARP.1	FAU_SAA.1	3
2	FAU_GEN.1	FPT_STM.1	Rationale (1)
3	FAU_SAA.1	FAU_GEN.1	2
4	FAU_SAR.1	FAU_GEN.1	2
5	FAU_SAR.3	FAU_SAR.1	4
6	FAU_STG.3	FAU_STG.1	Rationale (2)
7	FAU_STG.4	FAU_STG.1	Rationale (2)
8	FCS_CKM.1(1)	FCS_COP.1	12
		FCS_CKM.4	11
9	FCS_CKM.1(2)	FCS_COP.1	13
		FCS_CKM.4	11
10	FCS_CKM.2	FCS_CKM.1	8,9
		FCS_CKM.4	11
11	FCS_CKM.4	FCS_CKM.1	8, 9
12	FCS_COP.1(1)	FCS_CKM.1	8
		FCS_CKM.4	11
13	FCS_COP.1(2)	FCS_CKM.1	9
		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.2	20
18	FIA_IMA.1	-	-
19	FIA_SOS.1	-	-
20	FIA_UAU.2	FIA_UID.2	23
21	FIA_UAU.4	-	-

No.	Security Function Component	Dependencies	Reference No.
22	FIA_UAU.7	FIA_UAU.2	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.2	23
29	FPT_ITT.1	-	-
30	FPT_PST.1	-	-
31	FPT_TST.1	-	-
32	FPT_TEE.1	-	-
33	FTA_MCS.2	FIA_UID.2	23
34	FTA_SSL.5	FIA_UAU.2	20
35	FTA_TSE.1	-	-

[Table 5-11] 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 6-2].

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, CLIENT, programs, log messages	Audit records related to the security functions provided by the TOE, such as user data encryption and decryption
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

Component ID	Auditable Event
FAU_ARP.1	Response actions taken due to potential security breaches
FAU_SAA.1	Start and stop of operation of analysis mechanism, automatic response based on tools
FAU_STG.3	Response actions when the threshold is exceeded
FAU_STG.4	Response actions when audit storage fails
FCS_CKM.1(1)	Success and failure of actions
FCS_CKM.1(2)	Success and failure of actions
FCS_CKM.2	Success and failure of actions (Applies only to key distribution related to TSF data encryption and decryption)
FCS_CKM.4	Success or failure of the activity (Applicable only to key destruction related to encryption/decryption of the TSF data)
FCS_COP.1(1)	Success and failure of cryptographic operation, types of cryptographic operation
FCS_COP.1(2)	Success and failure of cryptographic operation, types of cryptographic operation
FDP_UDE.1	Success and failure in user data encryption/decryption
FIA_AFL.1	Threshold for failed authentication attempts reached, actions taken, and, if appropriate, subsequent restoration to normal state

Component ID	Auditable Event
FIA_IMA.1	Success or failure of mutual authentication Change in the authentication protocol
FIA_UAU.2	Any use of authentication mechanism
FIA_UAU.4	Attempts to reuse authentication data
FIA_UID.1	Any use of administrator identification mechanism including the provided administrator identity
FMT_MOF.1	All modifications to the functions in the TSF
FMT_MTD.1	All modifications to the values of the TSF data
FMT_PWD.1	All modifications to the password
FMT_SMF.1	Use of management functions
FMT_SMR.1	Modifications to user groups sharing roles
FPT_TST.1	Execution of the TSF self-tests and results of the tests
FPT_TEE.1	Execution of external entity test and test results
FTA_MCS.2	Denial of a new session based on the restrictions on the number of concurrent sessions
FTA_SSL.5	Locking or termination of an interactive session
FTA_TSE.1	Denial of session establishment based on the session establishment mechanism Any attempt to establish a user session

[Table 6-2] Auditable events

6.1.3. Analysis of and Response to Potential Violations

The TOE shall analyze potential violations based on the audit events and perform predefined actions accordingly.

Audit Events or Potential Violations	Rules	Actions
Authentication failure audit event among auditable events under FIA_UAU.1	Cumulative event count: 1	Notify authorized administrators via email
Integrity violation audit events and self-test failure of the validated cryptographic module among auditable events under FPT_TST.1	Cumulative event count: 1	Notify authorized administrators via email
Possible audit data loss specified in FAU_STG.3	When the audit event occurs	Notify authorized administrators via email
More than 5 consecutive failed authentication attempts as specified in FIA_AFL.1	Cumulative event count: 1	Notify authorized administrators via email
Concurrent administrator login attempt using the same account as specified in FTA_MCS.2	Cumulative event count: 1	Notify authorized administrators via email
Administrative access attempt from an unregistered IP and concurrent login attempt of the administrator having the same privilege as specified in FTA_TSE.1	Cumulative event count: 1	Notify authorized administrators via email

[Table 6-3] Audit events and 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. Action in case of possible audit data loss and prevention of audit data loss

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 V2.8 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 v2.0.1.1” in [Table 6-4] 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 6-3] below.

Purpose	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
TSF data encryption	Cryptographic key for the "User data encryption key" (Master key)	ISO/IEC 18031	Hash_DRBG (SHA256)	128bit
	Cryptographic key for the "Master key"	ISO/IEC 18031	Hash_DRBG (SHA256)	128bit
	API/Plugin configuration file Cryptographic key	ISO/IEC 18031	Hash_DRBG (SHA256)	128bit
	Cryptographic key for data transfer (Session key)	ISO/IEC 18031	Hash_DRBG (SHA256)	128bit
Mutual	API private key	ISO/IEC 18033-2	RSAES (SHA256)	2048bit

Purpose	Cryptographic Key	List of Standards	Cryptographic Key Generation Algorithm	Cryptographic Key Size
authentication	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 6-3] Cryptographic keys and their generation algorithms

Division	Details
Cryptographic Module Name	XecureCrypto 2.0.1.1
Validation Number	CM-153-2024.5
Validation Level	VSL1
Developer	HANCOM SECURE Inc.
Validation Date	May 2, 2019

[Table 6-4] 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].

Cryptographic Operation	Standards	Cryptographic Algorithm	Cryptographic Key Size
Block Cipher	KS X 1213	ARIA128	128 bits
		ARIA192	192 bits
		ARIA256	256 bits
	TTAS.KO-12.004/R1	SEED	128 bits

Cryptographic Operation	Standards	Cryptographic Algorithm	Cryptographic Key Size
Hash	ISO/IEC_10118-3	SHA224	224bits
		SHA256	256 bits
		SHA384	384 bits
		SHA512	512 bits

[Table 6-5] 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 Cipher	KS X 1213	ARIA128	128 bits
Hash	ISO/IEC_10118-3	SHA256	256 bits

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

6.2.3. Cryptographic key distribution

The user data encryption key is transmitted to a cryptographic channel (RSAES-OAEP, ARIA) formed after mutual authentication through a specified digital signature [RSA_PSS] that conforms to [ISO/IEC 18033-2].

Cryptographic Key	Timing of Distribution	Cryptographic Algorithm
User data encryption key	When key is requested by APIAgent/PluginAgent	The user data encryption key will be sent through a cryptographic channel authenticated by mutual authentication

[Table 6-7] Timing and method of distribution of the user data encryption key

6.2.4. 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.

Cryptographic Key	Timing of Deletion	Cryptographic Key Destruction Method
User data encryption key	APIAgent/PluginAgent : When encryption/decryption is finished Policy Server : When the "Discard" button in the "Cryptographic Key Management" menu is run	APIAgent/PluginAgent : Cryptographic keys are destroyed by overwriting the cryptographic key data with 0x00. Policy Server : Deleted from the cryptographic key storage
Cryptographic key for the "User data encryption key" (Master key)	Policy Server : When the "Discard" button in the "Cryptographic Key Management" menu is run	Deleted from the cryptographic key storage
Cryptographic key for data transfer (Session key)	When encryption/decryption for data transfer is finished	Cryptographic keys are destroyed by overwriting the cryptographic key data with 0x00.

[Table 6-8] Cryptographic key destruction

6.2.5. Random Bit Generation

The TOE generates random bits for cryptographic key generation using the HASH_DRBG (256bit) algorithm through the random bit generator of “XecureCrypto v2.0.1.1,” a validated cryptographic module whose safety and implementation conformity have been verified by the Korea Cryptographic Module Validation Program (KCMVP).

General information on the validated cryptographic module is provided in [Table 6-4].

※ Related security functional requirements

FCS_CKM.1, FCS_CKM.2, 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

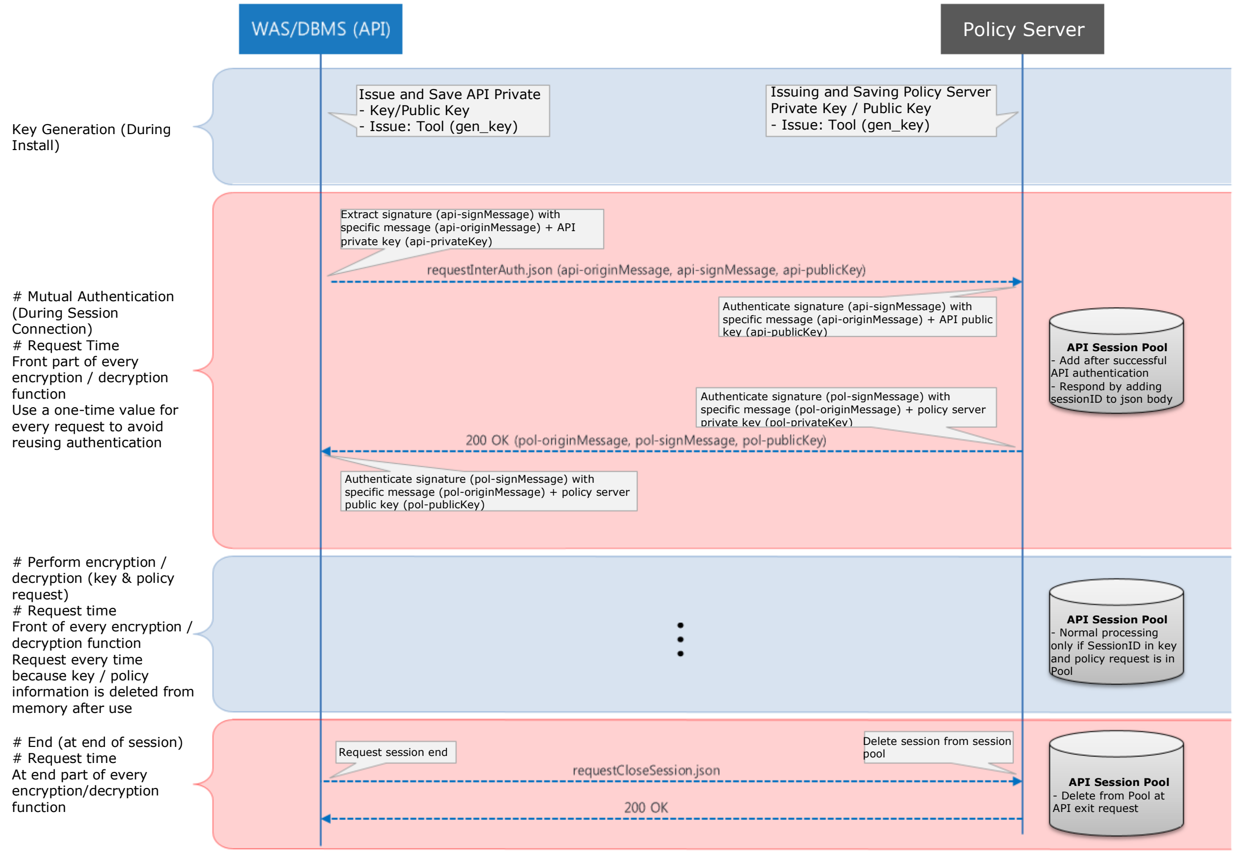
The TOE carries out the identification and authentication of authorized administrators through password verification.

6.4.1. Authentication failure handling

If 5 consecutive authentications fail during identification and authentication for the authorized administrator’s administrative access, it blocks administrative access to the account for 5 minutes and stores an audit record of the authentication failure.

6.4.2. Mutual authentication

Mutual authentication between APIAgent, PluginAgent, and Policy Server shall be carried out via the self-implemented authentication protocol. The method of mutual authentication is shown in [Figure 6-1].


WAS/DBMS (API)
Policy Server
Key Generation (During Install)
Issue and Save API Private - Key/Public Key - Issue: Tool (gen_key)
Issuing and Saving Policy Server Private Key / Public Key - Issue: Tool (gen_key)
# Mutual Authentication (During Session Connection) # Request Time Front part of every encryption / decryption function Use a one-time value for every request to avoid reusing authentication
Extract signature (api-signMessage) with specific message (api-originMessage) + API private key (api-privateKey)
requestInterAuth.json (api-originMessage, api-signMessage, api-publicKey)
Authenticate signature (api-signMessage) with specific message (api-originMessage) + API public key (api-publicKey)
API Session Pool - Add after successful API authentication - Respond by adding sessionID to json body
Authenticate signature (pol-signMessage) with specific message (pol-originMessage) + policy server private key (pol-privateKey)
200 OK (pol-originMessage, pol-signMessage, pol-publicKey)
Authenticate signature (pol-signMessage) with specific message (pol-originMessage) + policy server public key (pol-publicKey)
# Perform encryption / decryption (key & policy request) # Request time Front of every encryption / decryption function Request every time because key / policy information is deleted from memory after use
API Session Pool - Normal processing only if SessionID in key and policy request is in Pool
# End (at end of session) # Request time At end part of every encryption/decryption function
Request session end
requestCloseSession.json
Delete session from session pool
200 OK
API Session Pool - Delete from Pool at API exit request

[Figure 6-1] Mutual authentication flow between APIAgent, PluginAgent, and Policy Server

6.4.3. Password Policy Validation

A password value will be verified according to the password combination rules when creating and changing a security administrator's password.

The following verification mechanism is provided when creating a password:

- 9 to 20 characters with a combination of uppercase and lowercase letters, numbers, and special characters
- Uppercase letters: A–Z (26)
- Lowercase letters: a–z (26)
- Numbers: 0–9 (10)
- Special characters:!, @, #,%, ^,*, (,), -, =, _, +, [,], {, }; ,., /, >, ?

6.4.4. 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.

Random bits and hash values are generated using the validated cryptographic module "XecureCrypto v.2.0.1.1." The generated hash value is managed in the memory after login, and the integrity value will be checked upon login to prevent reuse of 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

Security administrators must perform security management functions through the security management interface policy server and can use the functions only after completing the identification and authentication process first. 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:

Security Function	Management Behavior
Policy Management - Policy	Query
	New
	Modify
	Delete

[Table 6-9] List of security functions

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 6-10].

TSF Data List	Management
Administrator's password	Modify
API/Plugin user data	Query
	New
	Modify
	Delete
Allowed IPs for administrator logins	Query
	New
	Modify
	Delete
Cryptographic key group info	Query
	New
	Modify
	Delete
Cryptographic key info	Query
	New
	Modify
	Delete
Encryption rule info	Query
	New
	Modify
	Delete
Encryption policy info	Query
	New
	Modify
	Delete
API/Plugin Installation IP	Query
	New
	Delete
Cryptographic key for the "Master Key"	Create
Cryptographic key for the "Environment configuration file"	Create
Key pair for mutual authentication	Create

[Table 6-10] 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 Roles

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.2.0.1.1," 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. Hancom xDB V2.8 APIAgent or PluginAgent installs the private key (PKCS#8 standard) and public key created at the Policy Server at the time of installation.
2. When PluginAgent or APIAgent sends the TSF data, APIAgent generates a random value using the KCMVP-validated cryptographic module (Session Key: 16byte Random: HASH_DRBG). The created random value will be used for TSF data encryption (ARIA 128, CBC mode), and checksum (SHA256) will be generated for the TSF data prior to encryption. The random value will then be encrypted (RSA-OAEP, 2048) with the public key, and a message (TSF data) created by combining the random value, encrypted TSF data, and checksum value will be sent.
3. The policy server decrypts (RSA-OAEP, 2048) the random value with the private key (PKCS#8, standard) after receiving the request message (TSF data), and then decrypts (ARIA 128, CBC mode) the request message with the decrypted random value.
4. Integrity verification is performed on the decrypted TSF data by verifying the checksum (SHA256), and the tasks (mutual authentication, user data key distribution) for the request will be carried out.
5. The policy server generates a random value using the KCMVP-validated cryptographic module (Session Key: 16byte Random: HASH_DRBG). The create random value will be used to encrypt (ARIA 128, CBC mode) the response message (TSF data), and checksum (SHA256) will be generated for the response message (TSF data) prior to encryption. The random value will then be encrypted (RSA-OAEP, 2048) with the public key, and a response message (TSF data) created by combining the random value, encrypted TSF data, and checksum value will be sent to APIAgent or PluginAgent. .
6. If the integrity verification fails, an error message is generated, and a random value is generated using the KCMVP-validated cryptographic module (Session Key: 16byte Random: HASH_DRBG). The created random value will be used to encrypt (ARIA 128, CBC mode) the error message, and checksum (SHA256) will be generated for the error

message prior to encryption. The random value will then be encrypted (RSA-OAEP, 2048) with the public key, and an error message (TSF data) created by combining the random value, encrypted TSF data, and checksum value will be sent to APIAgent or PluginAgent. .

7. APIAgent or PluginAgent decrypts (RSA-OAEP, 2048) the random value with the private key (PKCS#8, standard) after receiving the response message (TSF data), and then decrypts (ARIA 128, CBC mode) the request message (TSF data) with the decrypted random value.
8. Integrity verification is performed on the decrypted TSF data by verifying the checksum (SHA256), and the information received from the policy server will be used.

6.6.2. Basic Protection of Stored TSF data

The TOE encrypts, stores, and manages the target TSF data to protect the stored TSF data from unauthorized disclosure or modification.

Information requiring encryption includes administrator password and TOE configuration values (DB storage information and configuration file information). Administrator passwords are encrypted with SHA256, and TOE configuration values are encrypted with ARIA-CBC 128bit.

TOE configuration values are contained in the following TOE components: Hancom xDB V2.8 Policy Server, Hancom xDB V2.8 APIAgent, and Hancom xDB V2.8 PluginAgent.

Information requiring encryption, among the configuration file information of Hancom xDB V2.8 Policy Server, includes 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. Information requiring encryption among those managed in the Policy Server database includes administrator password, master key, and user data encryption key information.

Encryption target information among the configuration file information of Hancom xDB V2.8 APIAgent includes 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.

Encryption target information among the configuration file information of Hancom xDB V2.8 PluginAgent includes 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 TSF data encryption key for TSF data protection encrypts the TOE configuration values (ARIA-CBC 128bit).

The TSF data encryption key that protects TSF data is securely encrypted (ARIA-CBC 128bit) and protected.

The TSF data encryption key and master key are generated by the secure validated module "XecureCrypto 2.0.1.1."

The TSF data encryption key for encrypting TOE configuration values before product installation is generated through the validated module, and the cryptographic key for the "master key" used for protecting the KEK (master key) is also generated through the validated module and saved in the file.

To encrypt the TSF data of Hancom xDB V2.8 Policy Server, the "master key" is encrypted with the "cryptographic key for the master key" and "user data encryption key" with the "master key" before they are saved in the policy DB.

To encrypt the TSF data of Hancom xDB V2.8 APIAgent or Hancom xDB V2.8 PluginAgent, the TSF data encryption key is retrieved from the file, and the TOE configuration values are encrypted with the key and saved in the configuration file.

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

TSF Data		Applied Algorithm and Data	Encryption Requirements
Administrator's password		SHA256(Password+salt)	Requires encryption
TOE Settings	Policy Server public key path API public key path API private key path API private key password API user ID Policy Server IP Info Policy Server Port info	ARIA-CBC(data)	Requires encryption
TSF data cryptographic key	Cryptographic key for the "User data encryption key" (Master key)	ARIA-CBC(key)	Requires encryption
	Cryptographic key for the "Master key"	Operation Encoding	-
	API/Plugin configuration file Cryptographic key	ARIA-CBC(key)	Requires encryption

[Table 6-10] Protected TSF data and applied cryptographic algorithms

6.6.3. Self-test

The TOE runs a self-test periodically during normal operation and at start-up to verify the correct operation of the components (PolicyServer, PluginAgent, and APIAgent). The key processes for performing the TSF are subject to the self-test, and the self-test results of the validated cryptographic module are also received to point out any potential violations.

The TOE provides the function of verifying the integrity of TSF and the TSF data such as key executable files and configuration files. Monitoring will be performed at startup and periodically as desired by authorized administrators during normal operations. If an integrity violation is found as a result of the monitoring, the authorized administrator will be notified via e-mail.

TOE Components	Conditions for Invoking Monitoring
Hancom xDB V2.8 Policy Server	Perform integrity monitoring periodically during normal operation (daily) and at start-up
Hancom xDB V2.8 APIAgent	Perform integrity monitoring periodically during normal operation (daily) and at start-up
Hancom xDB V2.8 PluginAgent	Perform integrity monitoring periodically during normal operation (daily) and at start-up

[Table 6-11] Conditions for Invoking Monitoring

The TOE generates Sign values for the targets of process inspection and integrity check for the self-test at each designated inspection cycle and compares them with the stored Sign values (reference values) and includes the validated cryptographic module's self-test result as a target of the self-test. If an integrity violation is found, the TOE notifies the authorized administrator and generates audit data on the violation. The TOE performs monitoring on all environment configuration files such as security policy files necessary for the operation of the TOE as well as executable files. The TOE records audit data for the results of self-tests and integrity monitoring and the actions taken by the authorized administrators.

Purpose	Type	List of Standards	Cryptographic Key Generation Algorithm	Cryptographic Key Size
Self-test	Integrity Verification	ISO/IEC 18033-2	RSAES (SHA256)	2048

[Table 6-12] Algorithm used in Integrity Verification

TOE Type	Type	Name	Description
Hancom xDB V2.8 APIAgent	Library file	xecuredbapi.jar	Encryption/decryption library (Java)
		libxecuredbapi.so	Encryption/decryption library (C/C++)
		XDBAgent.jar	APIAgent core library
		libXecureASN.so	Module for ASN.1 operations
		libXecureCSP.so	Module that provides an interface between cryptographic libraries and other modules
		libXecureCodec.so	Module that provides a string conversion function
		libXecureIO.so	Module that provides functions related to memory, file, socket, time, etc.
		libXecurePKCS5.so	Password-based encryption
		libXecurePKCS8.so	Module that controls a user's secret key information
		libXecureSSL.so	Module that provides the functions required for SSL or TCPIP communication
		libXecureCrypto.so	Validated cryptographic module interface library
	Config file	xdf.ini	APIAgent environment configuration file
		config.json	APIAgent encryption environment 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	APIAgent TSF data cryptographic key
Hancom xDB V2.8 PluginAgent	Library file	xecuredbapi.jar	Encryption/decryption library (Java)
		libxecuredbapi.so	Encryption/decryption library (C/C++)
		XDBAgent.jar	PluginAgent core library
		libxdbplugin_comm_jni.so	Oracle Plugin library
		xdbplugin_comm_jni.class	Class for running Hancom xDB V2.8 API (Java) in Oracle
		libXecureASN.so libXecureASN.sl	Module for ASN.1 operations (OS dependable extensions)

TOE Type	Type	Name	Description
		libXecureCSP.so libXecureCSP.sl	Module that provides an interface between cryptographic libraries and other modules (OS dependable extensions)
		libXecureCodec.so libXecureCodec.sl	Module that provides a string conversion function (OS dependable extensions)
		libXecureIO.so libXecureIO.sl	Module that provides functions related to memory, file, socket, time, etc. (OS dependable extensions)
		libXecurePKCS5.so libXecurePKCS5.sl	Password-based encryption (OS dependable extensions)
		libXecurePKCS8.so libXecurePKCS8.sl	Module that controls a user's secret key information (OS dependable extensions)
		libXecureSSL.so libXecureSSL.so	Module that provides the functions required for SSL or TCPIP communication (OS dependable extensions)
		libXecureCrypto.so libXecureCrypto.sl	Validated cryptographic module interface library (OS dependable extensions)
	Config file	xdf.ini	APIAgent environment configuration file
		config.json	APIAgent encryption environment 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	APIAgent TSF data cryptographic key
Hancom xDB V2.8 Policy Server	The policy server shall perform integrity verification on all files under Hancom_xDB_V2.8/ in the installation path of the policy server. (Excluding Hancom_xDB_V2.8/logs)		

[Table 6-13] Integrity verification targets

6.6.4. External Entity Test

The TOE checks the connection status at initial startup to check whether the mail server (the external IT entity linked with the TOE) and the DBMS (the operating environment) are securely connected. The connection status of the DBMS will be checked as to whether a DB query can be executed successfully. If the connection status check fails, the process will be terminated.

TOE Components	**Timing of the Test**
Hancom xDB V2.8 Policy Server	Run external entity tests during initial start-up

[Table 6-14] Timing of external entity tests

External Entity	**External Entity Attribute**
DBMS	Whether a DB query was successfully executed
Mail server	Whether a test mail was successfully sent

[Table 6-15] External entities and their attributes

※ Related security functional requirements

FPT_ITT.1, FPT_PST.1, FPT_TST.1, FPT_TEE.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.5, FTA_TSE.1