SafeIdentity v5.1

Security Target(ST)

v1.2

The Security Target related to the certified TOE.

This Security Target is written in Korean and translated from Korean into English.

HANCOM WITH

Revision History

Document Version	Changes to Previous Revision	Revision Date	Author
v1.0	Initial registration of SafeIdentity v5.1 ST	09-12-2020	Changmin Park
V1.1	EOR-01 incorporated	08-13-2021	Changmin Park
V1.2	Security requirements incorporated	10-01-2021	Changmin Park

Table of Contents

1. ST Introduction

1.1. ST Reference

This ST is identified as follows:

Title	SafeIdentity v5.1 Security Target(ST)
Version	v1.2
Developer	IAM Development Team of Hancom WITH Inc.
Date	October 1, 2021
Evaluation Criteria	Common Criteria for Information Technology Security Evaluation
Common Criteria version	CC V3.1 r5
Evaluation Assurance Level	EAL1+ (ATE_FUN.1)
Protection Profile	Korean national protection profile for Single Sign On V1.1
Keywords	Single Sign On, SSO

1.2. TOE Reference

The Target of Evaluation (hereinafter referred to as the "TOE") that complies with this ST is identified as follows:

Classification	Specification		Version
TOE	SafeIdentity v5.1		5.1.02.211001
TOE Component	PolicyServer	SafeIdentity v5.1 PolicyServer	5.1.02.211001
	SafeAgent	SafeIdentity v5.1 SafeAgent	5.1.02.211001
Guidance Document	SafeIdentity v5.1 Operational Guidance(OPE)		v1.1
	SafeIdentity v5.1 Preparative Procedure(PRE)		v1.1

[Table 1-1] TOE Component and Version

- Team in charge of the development of SafeIdentity v5.1: IAM Development Team of Hancom WITH Inc.

1.3. TOE Overview

SafeIdentity v5.1 (hereinafter referred to as the "TOE") is used to enable a user to access various business systems to use services through a single login (Single Sign-On) without additional login actions. The TOE performs the user identification and authentication, the issuance of authentication tokens and the validity verification in accordance with the user authentication policies.

The TOE provides the user login capability using various ID/PW authentication methods, issues an authentication token during the user login, and verifies the issued authentication token if the user accesses another business system after the user login.

The primary security features provided by the TOE include the user identification and authentication, and the authentication token issuance/verification. The TOE uses a validated cryptographic module whose security and implementation conformance have been validated by the Korea Cryptographic Module Validation Program (KCMVP).

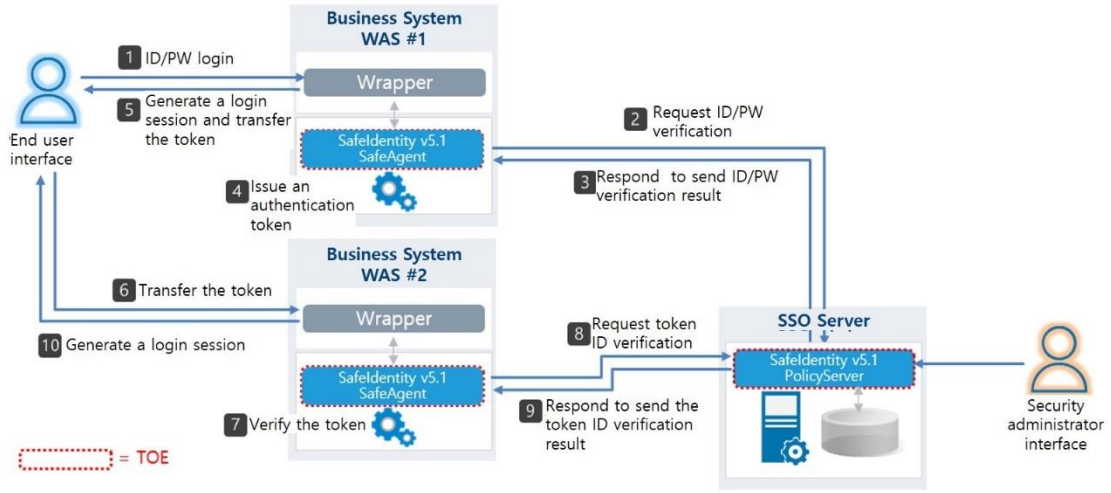
1.3.1. TOE usage and major security features

The TOE is a system that provides the function of Single Sign-On (SSO) that enables a user to access various business systems through a single login (SSO) without additional identification and authentication actions. A user who uses SSO offered by the TOE is free from the inconvenience of having to undergo separate authentication processes when using various business systems operated within an organization, which enhances the work efficiency. An administrator of the TOE can manage the security, and view audit records related to SSO by using the administrator interface provided by the TOE. The TOE provides the security audit function that manages major events by recording them as audit data when the security function and the management function are invoked; and the TSF protection function such as the protection of the data stored in the TSF-controlled repository and TSF self test. In addition, the TOE provides the identification and authentication function including authentication failure handling and TOE internal mutual authentication; the cryptographic support function such as cryptographic key management and cryptographic operation to issue authentication tokens; the security management function for the management of security functions behavior and configuration; and the TOE access function to manage access sessions of the authorized administrator. It also provides the function to verify the integrity of

major files such as main executable files and configuration files during the initial start-up and in a regular interval after the initial start-up.

The user identification and authentication procedure of the TOE is as shown in [Figure 1-1]. The user identification and authentication procedure can be divided into the initial authentication phase using ID/password, and the authentication token-based authentication phase that accesses the business system using the token issued during the initial authentication procedure. During the initial login, an end user requests login to the SSO business system by using ID/PW. The business system web transfers ID/PW to SafeIdentity v5.1 SafeAgent, a SSO agent (hereinafter "SafeAgent"), to request the login. SafeAgent that received the login request message sends a login verification request (ID/PW verification) to SafeIdentity v5.1 PolicyServer, a SSO server (hereinafter "PolicyServer"), which in turn checks the authorized user status. Upon receiving the login verification request, PolicyServer performs the login verification by using the user information stored in the DBMS. If the login verification result is valid, PolicyServer records the login history, and then responds to SafeAgent to send the login verification result. If the login is successfully verified, SafeAgent issues an authentication token, and transfers the authentication token to the end user.

The token-based authentication phase is performed only when the token has been normally issued in the initial authentication phase. When the user uses services in the business system, the issued token is transferred to SafeAgent installed in the pertinent business system, and SafeAgent that received the authentication token verifies it by interworking with PolicyServer. In this case, the encryption and decryption, the integrity and critical information of the authentication token are verified in SafeAgent and the validity of token ID value is verified in PolicyServer, which completes the verification of the authentication token.



[Figure 1-1] User identification and authentication procedure

[Table 1-2] below shows operations in each phase of the user identification and authentication procedure.

Authentication phase	Operation procedure
Initial authentication	1) D/PW login → 2) ID/PW verification request → 3) ID/PW verification result response → 4) Authentication token issuance → 5) Login session generation and authentication token transfer
Token-based authentication	6) Authentication token transfer → 7) Authentication token verification → 8) Token ID verification request → 9) Token ID verification result response → 10) Generation of login session

[Table 1-2] Operation procedure by authentication phase

In addition, a subject who issues, stores and verifies the authentication token is as follows:
- A subject who issues the authentication token: SafeAgent + PolicyServer
- Authentication token storage location: End user interface (web browser)
- A subject who verifies the authentication token: SafeAgent + PolicyServer

Major features of each component are as follows:

- PolicyServer

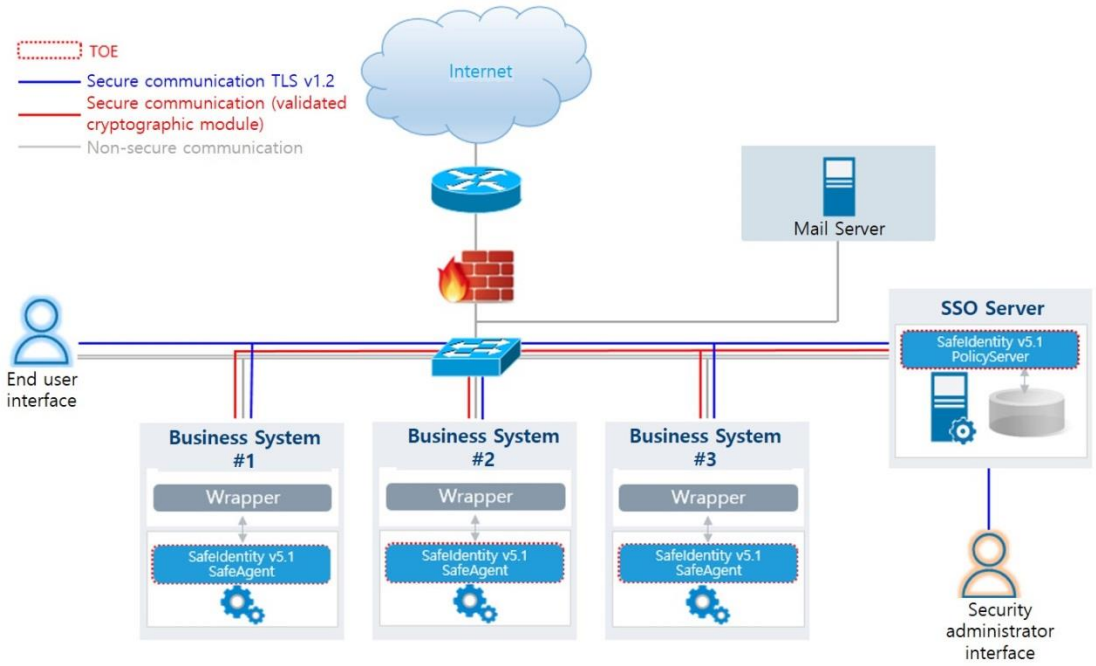
- User authentication information management server for Single Sign-On
- SSO and policy service, and encrypted communication with SafeAgent
- Prevents duplicated logins by maintaining authentication sessions in real time
- Receives log data from SafeAgent and stores them on the log server in real time
- Generates values necessary for issuing a token such as TokenID, idle time UUR and session slot information, and transfers them to SafeAgent
- Provides an administrator interface for the security management of the TOE
- Provides a function to register, modify and delete SSO user information
- Provides a function to manage users at an organizational level
- Provides the security audit function

• SafeAgent
  - A module that performs a function to authenticate a user transferred from Wrapper (initial authentication, token-based authentication)
  - Loaded on the login server and modifiable application server
  - Sends a request to authenticate user authentication data to PolicyServer
  - Encryption token generation and verification service, and encryption of user authentication data
  - Provides a programming interface to acquire and apply authentication data in a typical web environment

1.3.2. TOE type and operational environment

The TOE is a Single Sign-On (SSO) provided in the form of software that permits access to various business systems with a single login by a user.
The TOE consists of PolicyServer that processes user login, manages authentication tokens, and establishes the policy, and SafeAgent that is installed in each business system and performs the function of token issuance and verification. PolicyServer is a process type while SafeAgent is an API+ type.

The operational environment of the TOE is as shown in [Figure 1-2].



[Figure 1-2] Operational environment diagram

The TOE consists of SSO agent SafeAgent and SSO server PolicyServer. PolicyServer provides a function to verify end user login information by using the user information stored in the DBMS, and a function of the security management interface of the TOE. A security administrator can access PolicyServer and perform the security management. SafeAgent, installed and operated in each business system, sends a request to PolicyServer to verify user login information and TokenID, which are components of an authentication token, and issues and verifies authentication tokens. Wrapper provided for the end user login is out of the scope of the TOE.

External IT entities necessary for the operation of the TOE include a mail server to notify the administrator in case of possible audit data loss. The mail server, business systems and so forth, except for the TOE, fall under the operational environment of the TOE.
Encrypted communication of the TOE is divided into TOE internal communication and communication between the TOE and a user. For encrypted communication between PolicyServer and SafeAgent, encrypted communication is performed, using a validated cryptographic module. Communication between a web server, which is an operational environment, and a user (security administrator and end user) uses TLS v1.2 for encrypted communication.

1.3.3. Identification of non-TOE hardware/software

While additional hardware and software are necessary for the operation of the TOE, they stay out of the scope of the evaluation.
Hardware and software requirements for the operation of the TOE are as follows:

(1) Minimum system requirements for the operation of the TOE

TOE component	Type	Minimum requirements
SafeIdentity v5.1 PolicyServer	CPU	Intel ® Core™ i7 3.4 GHz or higher
	RAM	8 GB or higher
	HDD	100GB or more necessary for the TOE installation
	NIC	100/1000 Ethernet Card 1 port or more
SafeIdentity v5.1 SafeAgent	CPU	Intel ® Core™ i7 3.4 GHz or higher
	RAM	8 GB or higher
	HDD	100GB or more necessary for the TOE installation
	NIC	100/1000 Ethernet Card 1 port or more

[Table 1-2] Required hardware specifications of the operational environment of the TOE

(2) Minimum administrator system requirements for the security management

Type	Item	Minimum requirement

SW	Web Browser	Chrome 94.0

[Table 1-3] Required hardware and software specifications of the administrator system

(3) Non-TOE software necessary for the operation of the TOE

TOE component	S/W	Usage
SafeIdentity v5.1 PolicyServer	Solaris 10 (x86_64, 64bit)	OS to operate the TOE
	Oracle 12c Version 12.2.0.1.0	DBMS to store TSF data
	Java(JRE) 1.8.0_301	JRE to operate the TOE
	apache-tomcat-8.5.72	Web application server to provide the security management screen
SafeIdentity v5.1 SafeAgent	Solaris 10 (x86_64, 64bit)	OS to operate the TOE
	Java(JRE) 1.8.0_301	JRE to operate the TOE
	apache-tomcat-8.5.72	Web application server to operate the TOE and business systems

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

(4) External IT entities used except for the TOE

Type	Description
Mail server	Server to send emails to the authorized administrator

[Table 1-5] External IT entity

1.4. TOE Description

This chapter describes the scope and the boundary of the TOE.

1.4.1. Physical scope of the TOE

The physical scope and the boundary of the TOE include SafeAgent, which is a SSO agent; PolicyServer, which is SSO server; and the preparative procedure (PRE) for SafeIdentity v5.1 and the user operational guidance (OPE) for SafeIdentity v5.1, which are the guidance documents.

The TOE consists of software and relevant guidance documents, as detailed in [Table 1-6].

(1) TOE details

Classification	Specification		Type	Distribution
TOE	SafeIdentity v5.1 (Version Detail : 5.1.02.211001)		S/W	-
TOE Component	SSO Server	SafeIdentity v5.1 PolicyServer 5.1.02.211001 (safeidentity_v5.1_policyserver_5.1.02.211001.tar.gz)		CD
	SSO Agent	SafeIdentity v5.1 SafeAgent 5.1.02.211001 (safeidentity_v5.1_safeagent_5.1.02.211001.tar.gz)		
Guidance Document	Preparative Procedure	SafeIdentity v5.1 Preparative Procedure(PRE) v1.1 (SafeIdentity v5.1 Preparative Procedure(PRE) v1.1.pdf)	Document (PDF)	
	Operational Guidance	SafeIdentity v5.1 Operational Guidance(OPE) v1.1 (SafeIdentity v5.1 Operational Guidance(OPE) v1.1.pdf)		

[Table 1-6] Physical scope of the TOE

(2) Validated cryptographic module

A validated cryptographic module used in the TOE is as follows:

TOE	S/W	Usage
SafeIdentity v5.1 Policy Server	XecureCrypto v2.0.1.1	Validated cryptographic module for key generation, destruction and updating, and

		cryptographic operation
SafeIdentity v5.1 SafeAgent	XecureCrypto v2.0.1.1	Validated cryptographic module for key generation, destruction and updating, and cryptographic operation

[Table 1-7] General information on the validated cryptographic module

Details on the validate cryptographic module included in the TOE are as follows:

Classification	Description
Cryptographic module name	XecureCrypto v2.0.1.1
Validation number	CM-153-2024.5
Validated level	Security level 1
Developer	Hancom WITH Inc.
Validation date	May 8, 2020

[Table 1-8] Details on the validated cryptographic module

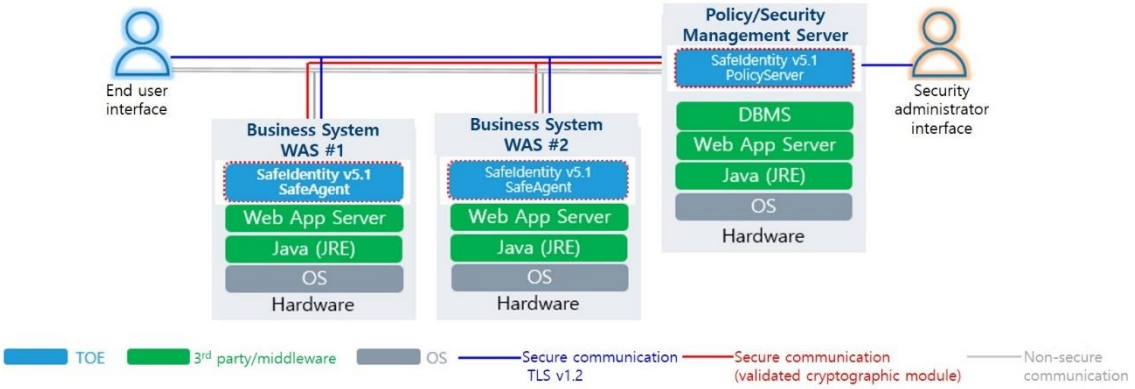
(3) Third-party software

Third-party software included in the TOE is as follows:

TOE	S/W	Version	Usage
SafeIdentity v5.1 Policy Server	Spring boot	2.5.5	Integrity verification
SafeIdentity v5.1 SafeAgent	Spring boot	2.5.5	Integrity verification
	H2 Database	1.4.200	Storage of integrity verification information

[Table 1-9] General information on third-party software

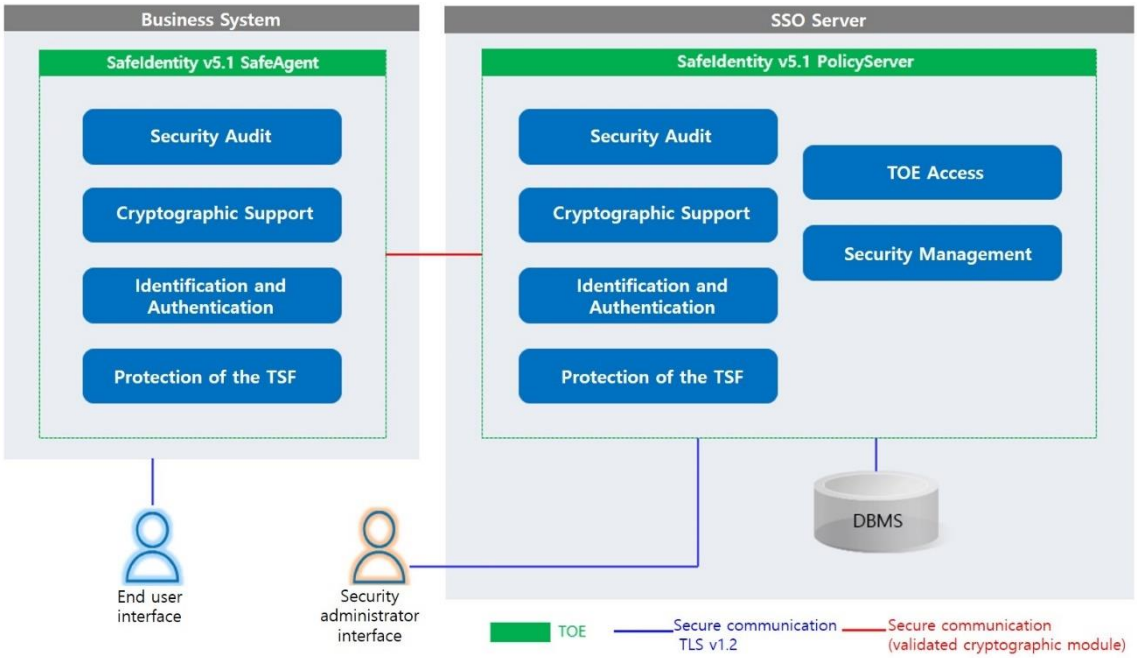
The physical scope of the TOE is as shown in [Figure 1-3].



[Figure 1-3] Physical scope of the TOE

Hardware platform where the TOE is installed, operating system, DBMS necessary for the operation of the TOE, web application server and so on are out of the scope of the TOE.

1.4.2. Logical scope of the TOE



[Figure 1-4] Logical scope of the TOE

(1) SafeAgent

- **Security Audit**

  The TOE generates and keeps track of audit records on auditable events, including behaviors of the security functions provided by the TOE and the security

management history. Information such as the time of occurrence of an auditable event, item, server IP, ID, access IP and outcome is recorded when audit records are generated. SafeAgent, a component to provide SSO, generates audit data on auditable event item, server IP, ID, access IP and outcome for each auditable event, and transmits audit data to PolicyServer, a SSO Server. With the audit data transmitted from SafeAgent, PolicyServer generates the time of occurrence, based on the system time, and stores the audit data in the DBMS.

- **Cryptographic Support**

  The TOE provides the function of cryptographic key generation, distribution and destruction, and cryptographic operation for the protection of data transmitted between the TOE components, encryption/decryption of user authentication tokens, and the protection of the stored data (TSF data). It also provides the function of random bit generation for the secure cryptographic key generation. The TOE performs the cryptographic key generation for authentication token encryption and TSF data encryption by using approved cryptographic algorithms of validated cryptographic module "XecureCrypto v2.0.1.1" whose security and implementation conformance have been validated by the KCMVP. If the use of the cryptographic support function is completed or the process is terminated, and thus, a cryptographic key is no longer needed, the cryptographic key is deleted, and then destroyed by means of zeroization to overwrite it with zeros, or by being overwritten with a new cryptographic key.

  The TOE performs cryptographic operations for user authentication token and data encryption by using approved cryptographic algorithms of validated cryptographic module "XecureCrypto v2.0.1.1" whose security and implementation conformance have been validated by the KCMVP. The validated cryptographic module is operated in the approved mode of operation during cryptographic operations. CBC mode is used when performing the encryption using the block cipher algorithm, and methods specified in KS X 1213 and TTAS.KO-12.004/R1 are applied in using IV. The TOE performs cryptographic operations for the encryption of TSF data, during which the validated cryptographic module is operated in the approved mode of operation.

  If the TOE uses random bits in SFRs that require the use of approved cryptographic algorithms of the validated cryptographic module, as in the case of the generation of major cryptographic keys including a key for user authentication token and data encryption (DEK), it uses a random bit generator of "XecureCrypto v2.0.1.1," a validated cryptographic module whose security and implementation conformance have been validated by the KCMVP.

A key for the data encryption for the encrypted communication is generated by using a random bit generator of validated cryptographic module "XecureCrypto v2.0.1.1." A session key and a session MAC key are encrypted and distributed through public key cryptography between the TOE components during the mutual authentication and upon the request of the authentication function.

For key exchange during the TOE internal mutual authentication, keys are distributed in accordance with a protocol developed by Hancom WITH Inc.

- **Identification and Authentication**

  The TOE performs the mutual authentication with PolicyServer, a TOE component, repetitively before it uses security functions provided by the TOE (except for self testing). The mutual authentication is performed through the process to generate and verify signatures between the TOE components by using approved cryptographic algorithms of validated cryptographic module "XecureCrypto v2.0.1.1."

  It also performs ID/PW-based authentication and token-based authentication through the process to generate and verify authentication tokens by using approved cryptographic algorithms of validated cryptographic module "XecureCrypto v2.0.1.1" in order to perform ID/PW-based authentication and token-based authentication of an end user.

  Random bits are generated with validated cryptographic module "XecureCrypto v2.0.1.1" and used as one-time authentication data (TokenID) in order to ensure the uniqueness of each token.

  A token issued is not stored in the TOE, but is destroyed in the memory after being transferred to a user browser. An authentication token is overwritten with zeros to be destroyed. When destroying a one-time authentication data (TokenID), the relevant memory is overwritten with zeros.

- **Protection of the TSF**

  The TOE protects the transmitted TSF data such as audit data and critical security parameters from disclosure and modification when it is transmitted between separate parts of the TOE by using approved cryptographic algorithms of validated cryptographic module "XecureCrypto v2.0.1.1" whose security and implementation conformance have been validated by the KCMVP.

  The TOE protects the authorized security administrator and end user passwords, cryptographic keys, critical security parameters, TOE configuration values (security policy, configuration parameter), audit data and so forth stored in the TSF data repository from unauthorized disclosure and modification. Especially, TSF data such

as the security administrator and end user passwords, data encryption key (DEK), critical security parameters, TOE configuration values and DBMS connection information are encrypted with approved cryptographic algorithms of validated cryptographic module "XecureCrypto v2.0.1.1," and then stored.

A data encryption key (DEK) is encrypted with approved cryptographic algorithms provided by the validated cryptographic module, using a key encryption key (KEK), and then stored.

Cryptographic keys and critical security parameters loaded onto the memory do not exist as plaintext in the memory at the time when the encryption/decryption operation is completed and thus, they are no longer needed.

The TOE is securely protected against an unauthorized attack from outside through the self-protection. Furthermore, it verifies the integrity of major files such as main executable files and configuration files of each TOE component during the initial start-up and in a regular interval, and performs self-validation of the cryptographic module and TOE functions.

(2) PolicyServer

- **Security Audit**

  The TOE generates and keeps track of audit records on auditable events, including behaviors of the security functions provided by the TOE and the security management history. Information such as the time of occurrence of an auditable event, item, server IP, ID, access IP and outcome is recorded when audit records are generated. PolicyServer, a component to provide SSO, generates audit data on auditable event, the implementation of the security management function, information on management access, etc. based on the initial authentication and the token-based authentication; stores the audit data in the DBMS, which is an operational environment; and provides the authorized administrator with the function of review or selective review of the audit data. In addition, the TOE analyzes potential violations using audit events, and takes pre-defined actions.

  The TOE provides the function to send an email to notify the authorized administrator if the audit trail exceeds a certain threshold. If the audit trail exceeds 90% of the total capacity defined, the administrator is notified via email. If it exceeds 90%, 10% is deleted and the administrator is notified via email.

- **Cryptographic Support**

The TOE provides the function of cryptographic key generation, distribution and destruction, and cryptographic operation for the protection of data transmitted between the TOE components, encryption/decryption of user authentication tokens, and the protection of the stored data (TSF data). It also provides the function of random bit generation for the secure cryptographic key generation. The TOE performs the cryptographic key generation for authentication token encryption and TSF data encryption by using approved cryptographic algorithms of validated cryptographic module "XecureCrypto v2.0.1.1" whose security and implementation conformance have been validated by the KCMVP. If the use of the cryptographic support function is completed or the process is terminated, and thus, a cryptographic key is no longer needed, the cryptographic key is deleted, and then destroyed by means of zeroization to overwrite with zeros, or by being overwritten with a new cryptographic key.

The TOE performs cryptographic operations for user authentication token and data encryption by using approved cryptographic algorithms of validated cryptographic module "XecureCrypto v2.0.1.1" whose security and implementation conformance have been validated by the KCMVP. The validated cryptographic module is operated in the approved mode of operation during cryptographic operations. CBC mode is used when performing the encryption using the block cipher algorithm, and methods specified in KS X 1213 and TTAS.KO-12.004/R1 are applied in using IV. The TOE performs cryptographic operations for the encryption of TSF data, during which the validated cryptographic module is operated in the approved mode of operation.

If the TOE uses random bits in SFRs that require the use of approved cryptographic algorithms of the validated cryptographic module, as in the case of the generation of major cryptographic keys including a key for user authentication token and data encryption (DEK), it uses a random bit generator of "XecureCrypto v2.0.1.1," a validated cryptographic module whose security and implementation conformance have been validated by the KCMVP.

- **Identification and Authentication**

  A user ID and password shall be verified during the initial authentication phase, which is a security function provided by the TOE, in order to provide the SSO service for the user.

  A password entered during the login in the initial authentication phase is checked to ensure that it is at least 9 up to 64 digits and mandatorily includes English alphabet, number and special character. If the authentication with the password fails for five times or more, the identification and authentication function becomes inactivated for

10 minutes so that the user cannot attempt to log in any longer. Once ID and password are verified, whether or not the initial authentication is successful is transferred to SafeAgent for the issuance of an authentication token.

The TOE performs the mutual authentication with SafeAgent, a TOE component, repetitively before it uses security functions provided by the TOE (except for self testing). The mutual authentication is performed through the process to generate and verify signatures between the TOE components by using approved cryptographic algorithms of validated cryptographic module "XecureCrypto v2.0.1.1."

The authorized administrator is identified and authentication before any action related to the security function in order to allow access to the security management function provided by the TOE.

The identity of the authorized security administrator is verified based on ID and password. A password generated must be at least 9 to 64 digits and include English alphabet, number and special character.

If the administrator fails to be authenticated for five times or more when attempting to log in to the security management screen, the identification and authentication function becomes inactivated for 10 minutes so that the administrator cannot log in any longer.

Passwords entered during the TOE security management login, or entered when adding, modifying end users or modifying administrator information are masked so that they are not displayed on the screen, in order to protect authentication feedback.

In addition, in case of failed identification and authentication, it does not provide feedback on the reason for the failure.

The TOE use a session ID to prevent the reuse of authentication data, and ensures the uniqueness of each authentication token by using TokenID issued by PolicyServer to prevent the reuse of authentication tokens.

- **TOE Access**

  The TOE controls access to the TOE by allowing only the registered IPs (up to 2) to access PolicyServer which is an interface for the security management. Accessible IPs are set during the TOE is installed. After the installation, it is possible to change accessible IPs through the security management screen.

  The TOE restricts the number of concurrent sessions to access the security management interface to 1 in order to block concurrent access to the same account. In case of multiple access attempts, the existing session is terminated and then new access is made, and the relevant audit log is generated.

If an end user or the authorized administrator remains inactive for 10 minutes after the login, the TOE terminates the session.

- **Security Management**

  The TOE provides the security management function that enables the authorized administrator to set and manage security policy, critical data and so forth. There is only one top administrator account for the authorized administrator. The authorized administrator can make settings to add, modify, delete SSO information of end users, and add, modify and delete an organization through PolicyServer. It is also possible to set the administrator information, and set accessible IPs for the security management access. The authorized administrator can change administrator or end user passwords through the interface. If an end user or the authorized administrator password is generated and modified, the validity of the password value is verified in accordance with the password policy. The TOE provides the function for an end user and the authorized administrator to change his/her password during the initial access.

- **Protection of the TSF**

  The TOE protects the transmitted TSF data such as audit data and critical security parameters from disclosure and modification when it is transmitted between separate parts of the TOE by using approved cryptographic algorithms of validated cryptographic module "XecureCrypto v2.0.1.1" whose security and implementation conformance have been validated by the KCMVP.

  The TOE protects the authorized security administrator and end user passwords, cryptographic keys, critical security parameters, TOE configuration values (security policy, configuration parameter), audit data and so forth stored in the TSF data repository from unauthorized disclosure and modification. Especially, TSF data such as the security administrator and end user passwords, data encryption key (DEK), critical security parameters, TOE set values and DBMS connection information are encrypted with approved cryptographic algorithms of validated cryptographic module "XecureCrypto v2.0.1.1," and then stored.

  A data encryption key (DEK) is encrypted with approved cryptographic algorithms provided by the validated cryptographic module, using a key encryption key (KEK), and then stored.

  Cryptographic keys and critical security parameters loaded onto the memory do not exist as plaintext in the memory at the time when the encryption/decryption operation is completed and thus, they are no longer needed.

The TOE is securely protected against an unauthorized attack from outside through the self-protection. Furthermore, it verifies the integrity of major files such as main executable files and configuration files of each TOE component during the initial start-up and in a regular interval, and performs self-validation of the cryptographic module and TOE functions.

1.5. Conventions

The notation, formatting and conventions used in this ST are consistent with the Common Criteria for Information Technology Security Evaluation.
The CC allows several operations to be performed for functional requirements: iteration, assignment, selection and refinement. Each operation is used in this ST.

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

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

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

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

1.6. Terms and Definitions

Technical terms in this ST are defined as follows. Terms used herein, which are the same as in the CC, must follow those in the CC.

▪ **Private Key**

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

▪ **Object**

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

▪ **Approved mode of operation**

Operation mode of a cryptographic module using an approved cryptographic algorithm

▪ **Approved cryptographic algorithm**

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

▪ **Validated Cryptographic Module**

A cryptographic module that is validated and given a validation number by the validation institution

▪ **Public Key**

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

▪ **Public Key (asymmetric) cryptographic algorithm**

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

▪ **Attack potential**

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

▪ **Management access**

Access attempts made by an administrator using HTTPS, SSH, TLS, etc. for the purpose of the management of the TOE

▪ **Management Console**

Application program that provides an administrator with graphic user interface (GUI) or command line interface (CLI) for system management, configuration and so forth

▪ **Recommend/be recommended**

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

▪ **Random bit generator (RBG)**

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

▪ **Symmetric cryptographic technique**

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

▪ **Iteration**

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

▪ **Security Target (ST)**

Implementation-dependent statement of security needs for a specified identified TOE

▪ **Security Policy Document**

Document uploaded to the list of the validated cryptographic module with the module's name and specifying the summary for the cryptographic algorithms and operational environments of the TOE

▪ **Protection Profile (PP)**

Implementation-independent statement of security needs for a TOE type

▪ **Decryption**

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

▪ **Secret Key**

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

▪ **User**

Refer to "External entity." User in the TOE means administrator and end user.

▪ **Selection**

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

▪ **Identity**

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

▪ **Encryption**

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

▪ **Korea Cryptographic Module Validation Program (KCMVP)**

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

▪ **Business System**

Application server that an authorized end user intends to access through Single-Sign On

▪ **Element**

Indivisible statement of a security need

▪ **Role**

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

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

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

▪ **Operation (on a subject)**

Specific type of action performed by a subject on an object

▪ **External entity**

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

▪ **Threat client**

Unauthorized external entity that adversely act on assets such as illegal access, modification or deletion

▪ **Authorized administrator**

Authorized user to securely operate and manage the TOE

▪ **Authorized user**

TOE user who may, in accordance with the Security Functional Requirements (SFRF), perform an operation

▪ **End user**

TOE user who wants to use the business system, not the administrator of the TOE

▪ **Authentication data**

Information used to verify a user's claimed identity

▪ **Authentication token**

Authentication data used for access by an authorized end user to a business system

▪ **Self-test**

Pre-operational or conditional test executed by the cryptographic module

▪ **Assets**

Entities that the owner of the TOE presumably places value upon

▪ **Refinement**

Addition of details to a component

▪ **Dependency**

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

▪ **Subject**

Active entity in the TOE that performs operations on objects

▪ **Sensitive Security Parameters (SSP)**

Critical security parameter (CSP) and public security parameter (PSP)

▪ **Augmentation**

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

▪ **Component**

Smallest selectable set of elements on which requirements may be based

▪ **Client**

Application program that can access SSO server or a client's service through a network

▪ **Class**

Set of CC families that share a common focus

▪ **Family**

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

▪ **Target of Evaluation (TOE)**

Set of software or hardware possibly accompanied by guidance

▪ **Evaluation Assurance Level (EAL)**

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

▪ **Can/could**

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

▪ **Assignment**

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

▪ **Shall/must**

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

▪ **Critical Security Parameters (CSP)**

Security-related information whose disclosure or modification can compromise the security of a cryptographic module (e.g., secret key/private key, password or authentication data such as PINs)

▪ **Application Programming Interface (API)**

A set of system libraries that exist between the application layer and the platform system and enables the easy development of the application running on the platform

▪ **Database Management System (DBMS)**

Software system that was built to configure and apply the database

▪ **Secure Sockets Layer (SSL)**

Security protocol proposed by Netscape in order to provide the security including confidentiality and integrity in a computer network

▪ **Transport Layer Security (TLS)**

Cryptographic protocol between a SSL-based server and a client, which is described in RFC 2246

▪ **TOE Security Functionality (TSF)**

Combined functionality of all hardware, software and firmware of a TOE that must be relied upon for the correct enforcement of the Security Functional Requirements (SFR)

▪ **TSF data**

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

▪ **Wrapper**

Interfaces for interconnection between the TOE and various types of business systems or authentication systems

2. Conformance Claims

This chapter describes how this ST conforms with the CC, the PP and the package.

2.1. CC Conformance Claim

CC		Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5 - Common Criteria for Information Technology Security Evaluation. Part 1: Introduction and General Model, Version 3.1, Revision 5 (CCMB-2017-04-001, April, 2017) - Common Criteria for Information Technology Security Evaluation. Part 2: Security Functional Components, Version 3.1, Revision 5 (CCMB-2017-04-002, April, 2017) - Common Criteria for Information Technology Security Evaluation. Part 3: Security Assurance Components, Version 3.1, Revision 5 (CCMB-2017-04-003, April, 2017)
Protection Profile		Korean national protection profile for Single Sign On V1.1
Conformance claim	**Part 2 Security functional components**	Extended : FCS_RBG.1, FIA_IMA.1, FIA_SOS.3, FMT_PWD.1, FPT_PST.1, FPT_TUD.1, FTA_SSL.5
	Part 3 Security assurance components	Conformant
	Package	Augmented : EAL1 augmented (ATE_FUN.1)

2.2. PP Conformance Claim

This ST complies with the security objectives and security requirements for the operating environment in the same way by strictly complying with the 'National Integrated Certification Protection Profile V1.1 (KECS-PP-0822a-2017)'

2.3. Package Conformance Claim

This ST claims conformance to assurance requirement package EAL1 and additionally defines some assurance requirements.

- Assurance Package: EAL1 augmented (ATE_FUN.1)

2.4. Conformance Claim Rationale

TOE Type, Operational Environment, Security Functional Requirements, Assurance Requirements Rationale for Compliance with PP

Classification	PP	ST	Rationale
TOE Type	Single Sign-On	Single Sign-On	Same as the PP
Operational Environment	OE.PHYSICAL_CONTROL	OE.PHYSICAL_CONTROL	Same as the PP
	OE.TRUSTED_ADMIN	OE.TRUSTED_ADMIN	Same as the PP
	OE.LOG_BACKUP	OE.LOG_BACKUP	Same as the PP
	OE.OPERATION_SYSTEM_REINFORCEMENT	OE.OPERATION_SYSTEM_REINFORCEMENT	Same as the PP
	OE.SECURE_DEVELOPMENT	OE.SECURE_DEVELOPMENT	Same as the PP
	-	OE.TIME_STAMP	Addition of FPT_STM.1 due to substitution of security objectives for the operational environment
	-	OE.DBMS	Addition of FAU_STG.1 due to substitution of security objectives for the operational environment
	-	OE.SECURE_CHANNEL	Addition of FTP_TRP.1 due to substitution of security objectives for the operational environment
Security Functional Requirements	FAU_ARP.1	FAU_ARP.1	Same as the PP
	FAU_GEN.1	FAU_GEN.1	Same as the PP
	FAU_SAA.1	FAU_SAA.1	Same as the PP
	FAU_SAR.1	FAU_SAR.1	Same as the PP
	FAU_SAR.3	FAU_SAR.3	Same as the PP
	FAU_STG.3	FAU_STG.3	Same as the PP
	FAU_STG.4	FAU_STG.4	Same as the PP
	FCS_CKM.1	FCS_CKM.1	Same as the PP

	FCS_CKM.2	FCS_CKM.2	Same as the PP
	FCS_CKM.4	FCS_CKM.4	Same as the PP
	FCS_COP.1	FCS_COP.1(1) FCS_COP.1(2) FCS_COP.1(3) FCS_COP.1(4) FCS_COP.1(5)	Apply iterative operation
	FCS_RBG.1	FCS_RBG.1	Same as the PP
	FIA_IMA.1	FIA_IMA.1	Same as the PP
	FIA_AFL.1	FIA_AFL.1	Same as the PP
	FIA_SOS.1	FIA_SOS.1	Same as the PP
	FIA_SOS.2	FIA_SOS.2	Same as the PP
	FIA_SOS.3	FIA_SOS.3	Same as the PP
	FIA_UAU.1	FIA_UAU.2	Application of SFR, which is a hierarchical relationship
	FIA_UAU.4	FIA_UAU.4	Same as the PP
	FIA_UAU.7	FIA_UAU.7	Same as the PP
	FIA_UID.1	FIA_UID.2	Application of SFR, which is a hierarchical relationship
	FMT_MOF.1	FMT_MOF.1	Same as the PP
	FMT_MTD.1	FMT_MTD.1	Same as the PP
	FMT_PWD.1	FMT_PWD.1	Same as the PP
	FMT_SMF.1	FMT_SMF.1	Same as the PP
	FMT_SMR.1	FMT_SMR.1	Same as the PP
	FPT_ITT.1	FPT_ITT.1	Same as the PP
	FPT_PST.1	FPT_PST.1	Same as the PP
	FPT_TST.1	FPT_TST.1	Same as the PP
	FTA_MCS.2	FTA_MCS.2	Same as the PP
	FTA_SSL.5	FTA_SSL.5	Same as the PP
	FTA_TSE.1	FTA_TSE.1	Same as the PP
Assurance Requirements	ADV_FSP.1	ADV_FSP.1	Same as the PP
	AGD_OPE.1	AGD_OPE.1	Same as the PP
	AGD_PRE.1	AGD_PRE.1	Same as the PP

	ALC_CMC.1	ALC_CMC.1	Same as the PP
	ALC_CMS.1	ALC_CMS.1	Same as the PP
	ASE_CCL.1	ASE_CCL.1	Same as the PP
	ASE_ECD.1	ASE_ECD.1	Same as the PP
	ASE_INT.1	ASE_INT.1	Same as the PP
	ASE_OBJ.1	ASE_OBJ.1	Same as the PP
	ASE_REQ.1	ASE_REQ.1	Same as the PP
	ASE_TSS.1	ASE_TSS.1	Same as the PP
	ATE_FUN.1	ATE_FUN.1	Same as the PP
	ATE_IND.1	ATE_IND.1	Same as the PP
	AVA_VAN.1	AVA_VAN.1	Same as the PP

[Table 2-1] TOE Type and Operational Environment, Security Requirements, Assurance Requirements Rationale

2.5. PP Conformance Statement

Since this ST adopts the TOE type, security objectives and security requirements in the same way as the Protection Profile, this ST "strictly conforms to the National Protection Profile for Single Sign-On V1.1."

3. Security Objectives

The followings are the security objectives handled by technical and procedural method supported from operational environment in order to provide the TOE security functionality accurately.

OE.PHYSICAL_CONTROL

The place where the SSO Agent and the SSO Server, among the TOE components, are installed and operated shall be equipped with access control and protection facilities so that it is accessible only by an authorized administrator.

OE.TRUSTED_ADMIN

The authorized administrator of the TOE shall not have malicious intentions, have been properly trained for the TOE management functions and shall accurately fulfill the duties in accordance with the administrator guidance.

OE.LOG_BACKUP

The authorized administrator shall check the spare space in the audit data repository on a periodic basis in preparation for audit record loss, and carry out audit data backup (external log server, separate storage device, etc.) to prevent audit data loss.

OE.OPERATION_SYSTEM_REINFORCEMENT

The authorized administrator of the TOE shall ensure the reliability and the security of the operating system by taking reinforcement measures for the operation system on which the TOE is installed and operated to address the latest vulnerabilities.

OE.SECURE_DEVELOPMENT

A developer who uses the TOE to interoperate with the user identification and authentication function in the operational environment of the business system shall ensure that the security functions of the TOE are securely applied in accordance with the requirements specified in the guidance document provided along with the TOE.

OE. SECURE_DBMS

Security policies and audit records stored in the TOE are stored in the database. The database shall not be generated, modified or deleted without a request from the TOE.

OE.TIME_STAMP

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

OE. SECURE PATH/CHANNEL

Any information transmitted while the authorized administrator accesses the management server via a web browser shall be protected through a secure path/channel.

4. Extended Components Definition

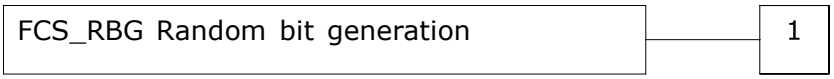
4.1. Cryptographic support

4.1.1. Random Bit Generation

Family Behaviour

This family defines requirements for the TSF to provide the capability that generates random bitsrequired for TOE cryptographic operation.

Component leveling



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

Management: FCS_RBG.1

There are no management activities foreseen.

Audit: FCS_RBG.1

There are no auditable events foreseen.

4.1.1.1. FCS_RBG.1 Random bit generation

Hierarchical to	No other components.
Dependencies	No dependencies.
FCS_RBG.1.1	The TSF shall generate random bits required to generate an cryptographickey using the specified random bit generator that meets the following [assignment: *list of standards*].

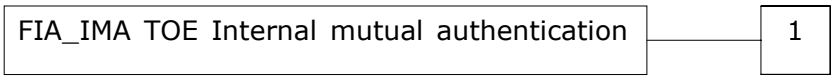
4.2. Identification and authentication

4.2.1. TOE Internal mutual authentication

Family Behaviour

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

Component leveling



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

Management: FIA_IMA.1

There are no management activities foreseen.

Audit: FIA_IMA.1

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

a) Minimum: Success and failure of mutual authentication

4.2.1.1. FIA_IMA.1 TOE Internal mutual authentication

Hierarchical to	No other components.
Dependencies	No dependencies.
FIA_IMA.1.1	The TSF shall perform mutual authentication between [assignment: *different parts of TOE*] using the [assignment: authentication protocol] that meets thefollowing [assignment: *list of standards*].

4.2.2. Specification of Secrets

Family Behaviour

This family defines requirements for mechanisms that enforce defined quality metrics on providedsecrets and generate secrets to satisfy the defined metric.

Component leveling

The specification of secrets family in CC Part 2 is composed of 2 components. It is nowcomposed of three components, since this PP adds one more component as below.

※ The description on two components included in CC Part 2 is omitted.

FIA_SOS.3 Destruction of secrets requires, that the secret information be destroyed according tothe specified destruction method, which can be based on the assigned standard.

Management: FIA_SOS.3

There are no management activities foreseen.

Audit: FIA_SOS.3

The following actions are recommended to record if FAU_GEN Security audit data generation is

included in the PP/ST:

a) Minimum : Success and failure of the activity

4.2.2.1. FIA_SOS.3 Destruction of Secrets

Hierarchical to	No other components.
Dependencies	FIA_SOS.2 TSF Generation of secrets

FIA_SOS.3.1 The TSF shall destroy secrets in accordance with a specified secrets destruction method [assignment: secret destruction method] that meets the following: [assignment: list of standards].

Application notes
o This SFR can be applied to the user's token.

4.3. Security Management

4.3.1. ID and password

Family Behaviour

This family defines the capability that is required to control ID and password management used

in the TOE, and set or modify ID and/or password by

authorized users.Component leveling

FMT_PWD ID and password	1

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

Management: FMT_PWD.1

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

Audit: FMT_PWD.1

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

a) Minimum: All changes of the password

4.3.1.1. FMT_PWD.1 Management of ID and password

Hierarchical to	No other components.
Dependencies	FMT_SMF.1 Specification of management functions
	FMT_SMR.1 Security roles
FMT_PWD.1.1	The TSF shall restrict the ability to manage the password of

	[assignment: *list of functions*] to [assignment: *the authorized identified roles*]. 1. [assignment: *password combination rules and/or length*] 2. [assignment: *other management such as management of specialcharacters unusable for password, etc.*]
FMT_PWD.1.2	The TSF shall restrict the ability to manage the ID of [assignment: *list of functions*] to [assignment: *the authorized identified roles*]. 1. [assignment: *ID combination rules and/or length*] 2. [assignment: other management such as management of specialcharacters unusable for ID, etc.]
FMT_PWD.1.3	The TSF shall provide the capability for [selection, choose one of: *setting ID and password when installing, setting password when installing, changing the ID and password when the authorized administrator accesses for the first time, changing the password when the authorized administrator accesses for the first time*].

Application notes

- o If the TOE does not provide the capability for managing the ID and password combination rules by authorized roles, etc., 'None.' may be specified in assignment operations ofFMT_PWD.1.1, FMT_PWD.1.2.
- o The ID and password combination rules that can be set by authorized roles may include minimum and maximum length setting, mixing rule setting involving English upper case/lower case/number/special characters, etc.

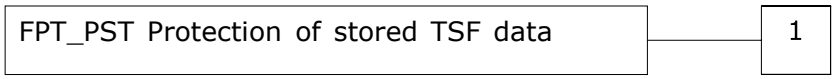
4.4. Protection of the TSF

4.4.1. Protection of stored TSF data

Family Behaviour

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

Component leveling



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

Management: FPT_PST.1

There are no management activities foreseen.

Audit: FPT_PST.1

There are no auditable events foreseen.

4.4.1.1. FPT_PST.1 Basic protection of stored TSF data

Hierarchical to	No other components.
Dependencies	No dependencies.
FPT_PST.1.1	The TSF shall protect [assignment: *TSF data*] stored in containers controlledby the TSF from the unauthorized [selection: *disclosure, modification*].

Application notes

- o Containers controlled by the TSF mean storage in the TOE or external entities (DBMS, etc.) that interact with the TOE.
- o Examples of TSF data to be protected as follows:
  - User password, cryptographic key (pre-shared key, symmetric key, private key, etc), TOE configuration values (security policy, environment setting, configuration parameters), audit data, etc.
- o The TSF data can be encrypted and stored to be protected from the unauthorized disclosure or modification.

4.4.2. TSF update

Family Behaviour

This family defines TOE firmware/software update requirements.

Component leveling

FPT_TUD.1 TSF security patch update, requires trusted update of the TOE firmware/softwareincluding the capability to verify the validity on the update file before installing updates.

Management: FPT_TUD.1

The following actions could be considered for the management functions in FMT:
a) Management of update file verification mechanism

Audit: FPT_TUD.1
The following actions are recommended to record if FAU_GEN Security audit data generation isincluded in the PP/ST:
a) Minimum: Update file verification result (success, failure)

4.4.2.1. FPT_TUD.1 TSF security patch update

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_TUD.1.1	The TSF shall provide the capability to view the TOE versions to [assignment: *the authorized identified roles*].
FPT_TUD.1.2	The TSF shall verify validity of the update files using [selection: hash valuecomparison, digital signature verification] before installing updates.

Application notes

- o The TSF shall provide the capability to check the current version of the TOE that mostrecently installed and executed by authorized roles.
- o The latest updates and security patches are essential to remove security vulnerabilities. Thevalidity verification on the update files is required since the installation of update files without any verification can result in system malfunction, or service failures, etc.

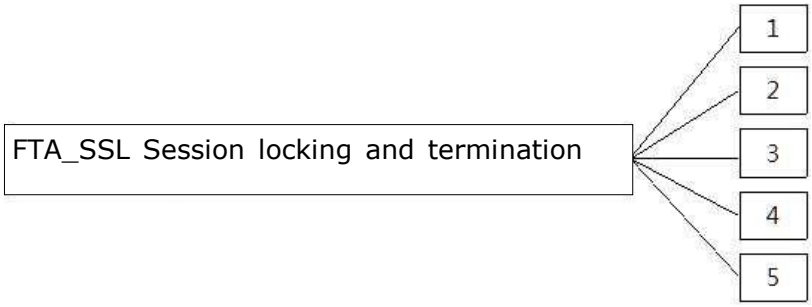
4.5. TOE Access

4.5.1. Session locking and termination

Family Behaviour

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

Component leveling



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

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

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

Management: FTA_SSL.5

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

a) Specification for the time interval of user inactivity that is occurred the session locking andtermination for each user

b) Specification for the time interval of default user inactivity that is occurred the session lockingand termination

Audit: FTA_SSL.5

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

a) Minimum: Locking or termination of interactive session

4.5.1.1. FTA_SSL.5 Management of TSF-initiated sessions

Hierarchical to	No other components.
Dependencies	FIA_UAU.1 authentication or No dependencies.
FTA_SSL.5.1	The TSF shall [selection: • lock the session and re-authenticate the user before unlocking thesession, • *terminate*] an interactive session after a [assignment: *time interval of userinactivity*].

Application notes

o This requirement can be applied to the management access of user(SSH, HTTPS, etc.).

5. Security Requirements

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

5.1. Security Functional Requirements

The security functional requirements included in this PP are derived from CC Part 2 and Chapter 4 Extended Components Definition.

Security functional class	Security functional component	
FAU	FAU_ARP.1	Security alarms
	FAU_GEN.1	Audit data generation
	FAU_SAA.1	Potential violation analysis
	FAU_SAR.1	Audit review
	FAU_SAR.3	Selectable audit review
	FAU_STG.3	Action in case of possible audit data loss
	FAU_STG.4	Prevention of audit data loss
FCS	FCS_CKM.1	Cryptographic key generation
	FCS_CKM.2	Cryptographic key distribution
	FCS_CKM.4	Cryptographic key destruction
	FCS_COP.1(1)	Cryptographic operation (Public Key)
	FCS_COP.1(2)	Cryptographic operation (MAC)
	FCS_COP.1(3)	Cryptographic operation (Hash)
	FCS_COP.1(4)	Cryptographic operation (Digital signature)
	FCS_COP.1(5)	Cryptographic operation (Symmetric key)
	FCS_RBG.1(Extended)	Random bit generation
FIA	FIA_AFL.1	Authentication failure handling
	FIA_IMA.1(Extended)	TOE Internal mutual authentication
	FIA_SOS.1	Verification of secrets
	FIA_SOS.2	TSF Generation of secrets
	FIA_SOS.3(Extended)	Destruction of secrets Mandatory
	FIA_UAU.2	User authentication before any action
	FIA_UAU.4	Single-use authentication mechanisms
	FIA_UAU.7	Protected authentication feedback

	FIA_UID.2	User identification before any action
FMT	FMT_MOF.1	Management of security functions behaviour
	FMT_MTD.1	Management of TSF data
	FMT_PWD.1(Extended)	Management of ID and password
	FMT_SMF.1	Specification of management functions
	FMT_SMR.1	Security roles
FPT	FPT_ITT.1	Basic internal TSF data transfer protection
	FPT_PST.1(Extended)	Basic protection of stored TSF data
	FPT_TST.1	TSF testing
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

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 take [Send a warning email to the authorized administrator] upon detection of a potentialsecurity 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 shut-down of the audit functions;

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

a) [Refer to the "auditable events" in [Table 5-2] Audit events, [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 applicable), and the outcome (success or failure) of the event; and
b) For each audit event type, based on the auditable event definitions of the functional components included in the ST [Refer to "Additional Audit Record" in [Table 5-2], [None]].

Security functional component	Auditable event	Additional audit record
FAU_ARP.1	Actions taken due to potential security violations	
FAU_SAA.1	Enabling and disabling of any of the analysis mechanisms, Automated responses performed by the tool	
FAU_STG.3	Actions taken due to exceeding of a threshold	
FAU_STG.4	Actions taken due to the audit storage failure	
FCS_CKM.1	Success and failure of the activity	
FCS_CKM.2	Success and failure of the activity (only applying to key distribution related to the TSF data encryption/decryption)	
FCS_CKM.4	Success and failure of the activity (only applying to key destruction related to the TSF data encryption/decryption)	
FCS_COP.1(1)	Success and failure, and the type of cryptographic operation(only applying to items related to the issue, storing, verification, and destruction of a token)	
FCS_COP.1(2)	Success and failure, and the type of cryptographic operation(only applying to items related to the issue, storing, verification, and destruction of a token)	
FCS_COP.1(3)	Success and failure, and the type of cryptographic operation(only applying to items related to the issue, storing, verification, and destruction of a token)	
FCS_COP.1(4)	Success and failure, and the type of cryptographic	

	operation(only applying to items related to the issue, storing, verification, and destruction of a token)	
FCS_COP.1(5)	Success and failure, and the type of cryptographic operation(only applying to items related to the issue, storing, verification, and destruction of a token)	
FIA_AFL.1	The reaching of the threshold for the unsuccessful authentication attempts and the actions taken, and the subsequent, if appropriate, restoration to the normal state	
FIA_SOS.2	Rejection by the TSF of any tested secret	
FIA_SOS.3(Extended)	Success and failure of the activity(applicable to the destruction of SSO token only)	
FIA_UAU.2	All use of the user authentication mechanism	
FIA_UAU.4	Attempts to reuse authentication data	
FIA_UID.2	All use of the user identification mechanism, including the administrator identity provided	
FMT_MOF.1	All modifications in the behaviour of the functions in the TSF	
FMT_MTD.1	All modifications to the values of TSF data	Modified values of TSF data
FMT_PWD.1(Extended)	All changes of the password	
FMT_SMF.1	Use of the management functions	
FMT_SMR.1	Modifications to the user group of rules divided	
FPT_TST.1	Execution of the TSF self tests and the results of the tests	Modified TSF data or execution code in case of integrity violation
FTA_MCS.2	Denial of a new session based on the limitation of multiple concurrent sessions	
FTA_SSL.5(Extended)	Locking or termination of interactive session	
FTA_TSE.1	Denial of a session establishment due to the session establishment mechanism	

[Table 5-2] Audit events

FAU_SAA.1 Potential violation analysis

Hierarchical to No other components.

Dependencies FAU_GEN.1 Audit data generation

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

FAU_SAA.1.2 The TSF shall enforce the following rules for monitoring audited evens:
a) Accumulation or combination of [
- Audit event of audit trail exceeding the threshold among auditable events in FAU_STG.3
- Audit event of full audit trail among auditable events in FAU_STG.4
- Audit event of failed authentication specified in FIA_UAU.2
- Audit event of integrity violation specified in FPT_TST.1
- Self test failure of the validated cryptographic module and self test (self validation) failure event specified in FPT_TST.1
- Attempts to access the security management from an unregistered IP and concurrent access to the same administrator account specified in FTA_TSE.1
] known to indicate a potential security violation;
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 administrator] with the capability to read [all the audit data] from the audit records.
FAU_SAR.1.2	The TSF shall provide the audit records in a manner suitable for the **authorized administrator** to interpret the information.

FAU_SAR.3 Selectable audit review

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

FAU_SAR.3.1 The TSF shall provide the ability to apply [descending order for time based on the result of the selected item] of audit data based on [AND operation of item, detailed item, server IP, ID, access IP, search period and outcome].

FAU_STG.3 Action in case of possible audit data loss

Hierarchical to Dependencies: No other components.
FAU_STG.1 Protected audit trail storage

FAU_STG.3.1 The TSF shall [notify the authorized administrator, [none]] if the audit trail exceeds [90% of the capacity allocated in the database].

FAU_STG.4 Prevention of audit data loss

Hierarchical to: FAU_STG.3 Action in case of possible audit data loss
Dependencies: FAU_STG.1 Protected audit trail storage

FAU_STG.4.1 The TSF shall *overwrite the oldest stored audit records* and [send an email to the authorized administrator to confirm whether or not to delete audit records] if the audit trail is full.

5.1.2. Cryptographic support (FCS)

FCS_CKM.1 Cryptographic key generation

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

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

Classification	Cryptographic key	List of Standards	Cryptographic Key Generation Algorithm	Cryptographic Key Sizes
Authentication token encryption	Authentication token encryption key (token cryptographic key)	ISO/IEC 18031	HASH_DRBG(SHA256)	128 bits
	Authentication token HMAC key	ISO/IEC	HASH_DRBG(SHA256)	128 bits

	(token MAC key)	18031		
	Authentication token cryptographic key encryption key (group key)	ISO/IEC 18031	HASH_DRBG(SHA256)	128 bits
TSF data encryption	TSF data encryption key (TSF cryptographic key)	ISO/IEC 18031	HASH_DRBG(SHA256)	128 bits
	TSF data HMAC key (TSF MAC key)	ISO/IEC 18031	HASH_DRBG(SHA256)	128 bits
	Secure channel session key (session key)	ISO/IEC 18031	HASH_DRBG(SHA256)	128 bits
	Secure channel session HMAC key (session MAC key)	ISO/IEC 18031	HASH_DRBG(SHA256)	128 bits
	SafeAgent private key	ISO/IEC 18033-2	RSAES(SHA-256)	2048 bits
	SafeAgent public key	ISO/IEC 18033-2	RSAES(SHA-256)	2048 bits
	PolicyServer private key	ISO/IEC 18033-2	RSAES(SHA-256)	2048 bits
	PolicyServer public key	ISO/IEC 18033-2	RSAES(SHA-256)	2048 bits
	private key encryption key(KEK)	ISO/IEC_9797-2	PBKDF2(HMAC-SHA-256)	128 bits

[Table 5-3] Cryptographic key generation algorithm

FCS_CKM.2 Cryptographic key distribution

Hierarchical to	No other components.
Dependencies	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction

FCS_CKM.2.1 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: [FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [cryptographic key destruction method in [Table 5-4]] that meets the following [none].

Classification of encryption	Cryptographic key	When deleted	Cryptographic key destruction method
Authentication token encryption	Authentication token encryption key (token cryptographic key)	Immediately after use	Overwrite with 0x00
	Authentication token HMAC key (token MAC key)	Immediately after use	Overwrite with 0x00
	Authentication token cryptographic key encryption key (group key)	Immediately after use	Overwrite with 0x00
TSF data encryption	TSF data encryption key (TSF cryptographic key)	Immediately after use	Overwrite with 0x00
	TSF data HMAC key (TSF MAC key)	Immediately after use	Overwrite with 0x00
	Secure channel session key (session key)	Immediately after use	Overwrite with 0x00
	Secure channel session HMAC key (session MAC key)	Immediately after use	Overwrite with 0x00
	SafeAgent private key	Immediately after use	Overwrite with 0x00
	SafeAgent public key	Immediately after use	Overwrite with 0x00

	PolicyServer private key	Immediately after use	Overwrite with 0x00
	PolicyServer public key	Immediately after use	Overwrite with 0x00
	private key encryption key(KEK)	Immediately after use	Overwrite with 0x00

[Table 5-4] Cryptographic key destruction method

FCS_COP.1(1) Cryptographic operation (Public key cryptographic operation)

Hierarchical to	No other components.
Dependencies	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1 The TSF shall perform [encryption and decryption] in accordance with a specified cryptographic algorithm [RSAES-OAEP] and a specified cryptographic key size [2048 bits] that meets the following [ISO/IEC 18033-2].

FCS_COP.1(2) Cryptographic operation (MAC)

Hierarchical to	No other components.
Dependencies	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1 The TSF shall perform [message authentication] in accordance with a specified cryptographic algorithm [HMAC-SHA-256] and a specified cryptographic key size [128 bits] that meets the following [ISO/IEC_9797-2].

FCS_COP.1(3) Cryptographic operation (Hash)

Hierarchical to	No other components.
Dependencies	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1 The TSF shall perform [hash] in accordance with a specified cryptographic algorithm [SHA-256] and a specified cryptographic key size [none] that meets the following [ISO/IEC_10118-3].

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

Hierarchical to	No other components.
Dependencies	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1 The TSF shall perform [digital signature] in accordance with a specified cryptographic algorithm [RSA-PSS] and a specified cryptographic key size [2048 bits] that meets the following [ISO/IEC 14888-2].

FCS_COP.1(5) Cryptographic operation (Symmetric key cryptographic operation)

Hierarchical to	No other components.
Dependencies	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1 The TSF shall perform [block cipher] in accordance with a specified cryptographic algorithm [ARIA_CBC, SEED_CBC] and a specified cryptographic key size [128 bits] that meets the following [KS X 1213, TTAS.KO-12.004/R1].

FCS_RBG.1 Random bit generation (Extended)

Hierarchical to	No other components.
Dependencies	No dependencies.

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

5.1.3. Identification and authentication (FIA)

FIA_AFL.1 Authentication failure handling

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

FIA_AFL.1.1 The TSF shall detect when *[5]* unsuccessful authentication attempts occur related to [end user and security administrator authentication attempt].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been *exceeded*, the TSF shall [prevent the account from being authentication for 10 minutes by default].

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 through [the authentication protocol developed by Hancom WITH Inc.] that meets [none] between [SafeAgent and PolicyServer].

FIA_SOS.1 Verification of secrets

Hierarchical to	No other components.
Dependencies	No dependencies.

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet [the combination rule to contain at least 3 types of letters among English alphabets, numbers and special characters, in at least 9 up to 64 digits].

FIA_SOS.2 TSF Generation of secrets

Hierarchical to	No other components.
Dependencies	No dependencies.

FIA_SOS.2.1 The TSF shall provide a mechanism to generate **an authentication tokens** that meet [authentication token generation steps specified in [Table 5-5]].

FIA_SOS.2.2 The TSF shall be able to enforce the use of TSF generated **authentication tokens** for [name of TSF in [Table 5-6]].

Token generation steps	Cryptographic algorithm	Cryptographic key size	Remarks
Decrypt a group key	SEED_CBC	128 bits	-
Generate a token cryptographic key and a token MAC key	HASH_DRBG - SHA256	128 bits	-
Encrypt the token cryptographic key and the token MAC key	SEED_CBC	128 bits	-
Encrypt the authentication token data	SEED_CBC	128 bits	-
Store additional information in the authentication token	-	-	-
Generate a message authentication code HMAC for the authentication token	HMAC-SHA256	128 bits	-

[Table 5-5] Authentication token generation steps and criteria

TSF name	Usage
Token-based authentication	Use of authentication tokens for SSO

[Table 5-6] List of TSFs using TSF generated authentication token

FIA_SOS.3 Destruction of secrets (Extended)

Hierarchical to No other components.
Dependencies FIA_SOS.2 TSF Generation of secrets

FIA_SOS.3.1 The TSF shall destroy **authentication tokens** in accordance with a specified **authentication token** destruction method [deletion token values stored in the browser memory, and zeroization of TokenID value in PolicyServer when a session is terminated] that meets the following [none].

FIA_UAU.2 User authentication before any action

Hierarchical to FIA_UAU.1 Timing of authentication.
Dependencies FIA_UID.1 Timing of identification

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

FIA_UAU.4 Single-use authentication mechanisms

Hierarchical to	No other components.
Dependencies	No dependencies.

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

FIA_UAU.7 Protected authentication feedback

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

FIA_UAU.7.1 The TSF shall provide only [information that masked the entered password with "*"] to the user while the authentication is in progress.

FIA_UID.2 Identification before any action

Hierarchical to	FIA_UID.1 Timing of identification.
Dependencies	No dependencies.

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

5.1.4. Security management (FMT)

FMT_MOF.1 Management of security functions behaviour

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

FMT_MOF.1.1 The TSF shall restrict the ability to conduct ***conduct management actions of*** the functions in [[Table 5-7] List of security functions] to [the authorized administrator].

Security function	Management action
End user management (Access Policy)	enable
	disable

[Table 5-7] List of management of security functions behavior

※ "Management action" to which a refinement operation is applied includes the ability to determine the behavior, disable, enable, modify the behavior of some functions in the TSF.

FMT_MTD.1 Management of TSF data

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

FMT_MTD.1.1 The TSF shall restrict the ability to ***manage*** [[Table 5-8] List of TSF data] to [the authorized administrator].

List of TSF data	Management action
Administrator information management	Query
	Modify
End user management	Query
	New(Add)
	Modify
	Delete
Organization management	Query
	New(Add)
	Modify

	Delete

[Table 5-8] List of TSF data

※ "Manage" to which a refinement operation is applied includes the ability to change default, query, modify, delete, clear, other operation, etc.

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 [the authorized administrator].

1. [none]
2. [none]

FMT_PWD.1.2 The TSF shall restrict the ability to manage ID of [none] to [the authorized a dministrator].

1. [none]
2. [none]

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

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 Timing of identification

FMT_SMR.1.1 The TSF shall maintain the role [security administrator].

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

5.1.5. 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 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 [

a) administrator and end user password
b) Authentication token's information
c) Cryptographic key
d) Critical security parameter
e) TOE set value (security policy, configuration parameter)
f) Audit data

] stored in the 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 a suite of self tests *during initial start-up, periodically during normal operation* to demonstrate the correct operation of *TSF*.

FPT_TST.1.2 The TSF shall provide **authorized administrator** with the capability to verify the integrity of *[[Table 5-9] TSF data]]*.

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

Classification	Type	Name	Description
SafeAgent	Verifies the integrity of all files located under SafeAgent / in the path where SafeAgent was installed.		
PolicyServer	Verifies the integrity of all files located under policyserver / in the path where PolicyServer was installed.		

[Table 5-9] TSF data

5.1.6. 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 Timing of identification

FTA_MCS.2.1 The TSF shall restrict the maximum number of concurrent sessions that belong to the same user according to the rules [restriction of the number of concurrent sessions of the management access by the administrator to 1,restriction of the number of concurrent sessions of the access by the end-user to 1, the rules on the maximum number of concurrent sessions {none}].

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

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

Hierarchical to No other components.
Dependencies FIA_UAU.1 Authentication or No dependencies.

FTA_SSL.5.1 The TSF shall *terminate* an interactive session after [10 minutes by default].

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 **administrator's management access session** establishment based on [access IP, *none*].

.

5.2. Security Assurance Requirements

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

Security assurance class	Security assurance component	
Security Target evaluation	ASE_INT.1	ST introduction
	ASE_CCL.1	Conformance claims
	ASE_OBJ.1	Security objectives for the operational environment
	ASE_ECD.1	Extended components definition
	ASE_REQ.1	Stated security requirements
	ASE_TSS.1	TOE summary specification
Development	ADV_FSP.1	Basic functional specification
Guidance documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative procedures
Life-cycle support	ALC_CMC.1	Labelling of the TOE
	ALC_CMS.1	TOE CM coverage
Tests	ATE_FUN.1	Functional testing
	ATE_IND.1	Independent testing : conformance
Vulnerability assessment	AVA_VAN.1	Vulnerability survey

[Table 5-10] Security assurance requirements

5.2.1. Security Target evaluation

ASE_INT.1 ST introduction

Dependencies No dependencies.

Developer action elements

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

Content and presentation elements

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

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

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

ASE_INT.1.4C The TOE overview shall summarise 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 elements

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

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

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

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

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

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

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

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

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

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

Evaluator action elements

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

ASE_OBJ.1 Security objectives for the operational environment

Dependencies No dependencies.

Developer action elements

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

Content and presentation elements

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

Evaluator action elements

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

ASE_ECD.1 Extended components definition

Dependencies No dependencies.

Developer action elements

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

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

Content and presentation elements

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

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

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

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

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

Evaluator action elements

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

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

ASE_REQ.1 Stated security requirements

Dependencies ASE_ECD.1 Extended components definition

Developer action elements

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

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

Content and presentation elements

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

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

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

ASE_REQ.1.4C All operations shall be performed correctly.

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

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

Evaluator action elements

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

ASE_TSS.1 TOE summary specification

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

Developer action elements

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

Content and presentation elements

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

Evaluator action elements

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

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

5.2.2. Development

ADV_FSP.1 Basic functional specification

Dependencies No dependencies.

Developer action elements

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

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

Content and presentation elements

ADV_FSP.1.1C The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.2C The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.3C The functional specification shall provide rationale for the implicit categorization of interfaces as SFR-non-interfering.

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

Evaluator action elements

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

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

5.2.3. Guidance documents

AGD_OPE.1 Operational user guidance

Dependencies ADV_FSP.1 Basic functional specification

Developer action elements

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

Content and presentation elements

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

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

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

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

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

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

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

Evaluator action elements

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

AGD_PRE.1 Preparative procedures

Dependencies No dependencies.

Developer action elements

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

Content and presentation elements

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

AGD_PRE1.2C The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

Evaluator action elements

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

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

5.2.4. Life-cycle support

ALC_CMC.1 Labelling of the TOE

Dependencies ALC_CMS.1 TOE CM coverage

Developer action elements

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

Content and presentation elements

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

Evaluator action elements

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

ALC_CMS.1 TOE CM coverage

Dependencies No dependencies.

Developer action elements

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

Content and presentation elements

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

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

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

5.2.5. Tests

ATE_FUN.1 Functional testing

Dependencies ATE_COV.1 Evidence of coverage

Developer action elements

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

ATE_FUN.1.2D The developer shall provide test documentation.

Content and presentation elements

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

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

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

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

Evaluator action elements

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

ATE_IND.1 Independent testing - conformance

Dependencies ADV_FSP.1 Basic functional specification
AGD_OPE.1 Operational user guidance
AGD_PRE.1 Preparative procedures

Developer action elements

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

Content and presentation elements

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

Evaluator action elements

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

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

5.2.6. Vulnerability assessment

AVA_VAN.1 Vulnerability survey

Dependencies ADV_FSP.1 Basic functional specification
AGD_OPE.1 Operational user guidance
AGD_PRE.1 Preparative procedures

Developer action elements

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

Content and presentation elements

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

Evaluator action elements

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

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

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

5.3. Security Requirements Rationale

5.3.1. Dependency of SFRs

The following table summarizes dependencies of the SFRs.

No.	Security functional requirements	Dependency	Reference No.
1	FAU_ARP.1	FAU_SAA.1	3
2	FAU_GEN.1	FPT_STM.1	*Rationale(1)
3	FAU_SAA.1	FAU_GEN.1	2
4	FAU_SAR.1	FAU_GEN.1	2
5	FAU_SAR.3	FAU_SAR.1	4
6	FAU_STG.3	FAU_STG.1	*Rationale(2)
7	FAU_STG.4	FAU_STG.1	*Rationale(2)
8	FCS_CKM.1	FCS_CKM.2 or FCS_COP.1	9, 11, 12, 13,

			14, 15
		FCS_CKM.4	10
9	FCS_CKM.2	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	8
		FCS_CKM.4	10
10	FCS_CKM.4	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	8
11	FCS_COP.1(1)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	8
		FCS_CKM.4	10
12	FCS_COP.1(2)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	8
		FCS_CKM.4	10
13	FCS_COP.1(3)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	*Rationale(3)
		FCS_CKM.4	*Rationale(3)
14	FCS_COP.1(4)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	8
		FCS_CKM.4	10
15	FCS_COP.1(5)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	8
		FCS_CKM.4	10
16	FCS_RBG.1	-	-
17	FIA_IMA.1	-	-
18	FIA_AFL.1	FIA_UAU.1	22 *Rationale(4)
19	FIA_SOS.1	-	-
20	FIA_SOS.2	-	-
21	FIA_SOS.3	FIA_SOS.2	20
22	FIA_UAU.2	FIA_UID.1	25 *Rationale(5)
23	FIA_UAU.4	-	-
24	FIA_UAU.7	FIA_UAU.1	22 *Rationale(4)
25	FIA_UID.2	-	-
26	FMT_MOF.1	FMT_SMF.1	29
		FMT_SMR.1	30
27	FMT_MTD.1	FMT_SMF.1	29
		FMT_SMR.1	30
28	FMT_PWD.1	FMT_SMF.1	29
		FMT_SMR.1	30
29	FMT_SMF.1	-	-
30	FMT_SMR.1	FIA_UID.1	25

			*Rationale(5)
31	FPT_ITT.1	-	-
32	FPT_PST.1	-	-
33	FPT_TST.1	-	-
34	FTA_MCS.2	FIA_UID.1	25 *Rationale(5)
35	FTA_SSL.5	FIA_UAU.1 or None	22 *Rationale(4)
36	FTA_TSE.1	-	-

[Table 5-11] Rationale for the dependency of the security functional requirements

Rationale (1): FAU_GEN.1 has a dependency on FPT_STM.1. However, FPT_STM.1 is satisfied by OE.TIME_STAMPS, which is a security objective for the operational environment, thereby satisfying the dependency.

Rationale (2): FAU_STG.3 and FAU_STG.4 have dependencies on FAU_STG.1. However, FAU_STG.1 is satisfied by OE.DBMS, which is a security objective for the operational environment, thereby satisfying the dependency.

Rationale (3): FCS_COP.1(3) has a dependency on FCS_CKM.1 and FCS_CKM.4. However, there is no cryptographic key generation and cryptographic key destruction due to natures of hash algorithms, thereby satisfying the dependency.

Rationale (4): FIA_AFL.1, FIA_UAU.7 and FTA_SSL.5 have dependencies on FIA_UAU.1. However, FIA_UAU.1 and FIA_UAU.2 are hierarchical, thereby satisfying the dependency.

Rationale (5): FIA_UAU.2, FMT_SMR.1 and FTA_MCS.2 have dependencies on FIA_UID.1. However, FIA_UID.1 and FIA_UID.2 are hierarchical, thereby satisfying the dependency.

5.3.2. Dependency rationale of SARs

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

The augmented SAR ATE_FUN.1 has dependency on ATE_COV.1. but, ATE_FUN.1 is augmented to require developer testing in order to check if the developer correctly

performed and documented the tests in the test documentation, ATE_COV.1 is not included in this PP since it is not necessarily required to show the correspondence between the tests and the TSFIs.

6. TOE Summary Specification

This chapter explains how the TOE meets security requirements in relation to the security functions of the TOE: security audit, cryptographic support, identification and authentication, security management, protection of the TSF and TOE access.

6.1 Security Audit

The TOE generates and stores audit data on security audit events that occur in each TOE component. Audit data include the date and time of an event, a type of the event, subject identity, an outcome of the event, etc. If a potential security violation is detected, it is notified to users in real time, and actions are taken according to a method specified for each type of security violation.

6.1.1. Security alarms

If the TOE detects a potential security violation from the generated audit data, it takes actions to respond to acts of security violation. Potential security violations are explained in [Table 6-3], and as a response, a security alarm email is sent to the authorized security administrator.

6.1.2. Audit data generation

The TOE generates and keeps track of audit records on auditable events, including behaviors of the security functions provided by the TOE and the security management history. Information such as the time of occurrence of an auditable event, item, server IP, ID, access IP and outcome is recorded when audit records are generated. SafeAgent and PolicyServer, which are components that provide SSO, generate audit data on auditable events that occur in the process of the initial authentication and token-based authentication, and audit data on the implementation of security management functions and management access information, and store them in the DBMS.

Classification	Field	Detail
Security audit data	Time, item, server IP, ID, access IP, outcome	Audit records related to the implementation of security functions provided by the TOE, audit records related to SSO performed by end users

TSF management data	Time, item, server IP, ID, access IP, outcome	Audit records on security management behaviors including the authorized administrator's login to the security management interface, and result of any addition, modification and deletion of TSF data by the authorized administrator

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

6.1.3. Potential violation analysis and response

The TOE analyzes potential violations based on audit events, and take pre-defined actions.

Potential violation audit event	Action
- A threshold of the audit log disk capacity is exceeded	If the audit log exceeds 90%, the authorized administrator is notified via email
- A audit log disk capacity is full	If the audit log exceeds 95%, 10% is deleted and the authorized administrator is notified via email
- Failed authentication attempts to log in for 5 consecutive times or more	Authentication is blocked for 10 minutes, and the authorized administrator is notified via email
- Audit event of integrity violation	The authorized administrator is notified via email
- Self test (self validation) failure	The authorized administrator is notified via email
- Concurrent access to the same administrator account	Previous access session is blocked, and the authorized administrator is notified via email
- Attempts to access the security management from an unregistered IP	Authentication is blocked, and the authorized administrator is notified via email

[Table 6-2] Potential violations and actions

6.1.4. Audit review and selectable audit review

After the authorized administrator logs on to PolicyServer, he/she can select the audit menu to perform the audit data review. The TOE provides the function of review and selective review of audit data generated by the TOE and stored in the DBMS. The administrator can review the audit data, in a descending order for time, for the result of the items selected based on AND operation of item, detailed items, server IP, ID, access IP, search period and outcome from the collected security audit data and TSF management data.

6.1.5. Action in case of possible audit data loss and prevention of audit data loss

The TOE protects the audit records stored in the audit trail from unauthorized modification and deletion. It does not provide the function to modify and delete any audit data through the administrator interface. In addition, the TOE provides the function to notify the authorized administrator via email if the audit trail exceeds a defined limit. If the audit trail exceeds 90% of the maximum defined capacity, it notifies the administrator via email, and if it exceeds 95%, it deletes 10% of the oldest audit data, and then notifies the administrator via email.

※ Relevant SFRs

- 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 is performed with symmetric key cryptographic operation. Cryptographic keys necessary for such encryption are generated with HASH_DRBG algorithms that meet the standard ISO/IEC 18031.

When generating cryptographic keys necessary for asymmetric key cryptographic operation, 2048-bit long cryptographic keys are generated with RSAES algorithms that meet the standard ISO/IEC 18033-2(2006).

Algorithms that encrypt an authentication token are encrypted through SEED-CBC (128 bits). An authentication key is used as one-time key that is generated upon the authentication and destroyed when the authentication expires.

A private key is encrypted with ARIA-CBC (128 bits) using a private key encryption key (KEK) derived from a password.

The TOE performs the cryptographic key generation using approved cryptographic algorithms of "XecureCrypto v2.0.1.1," a validated cryptographic module whose security and implementation conformance have been validated through the KCMVP. Cryptographic algorithm and cryptographic key size of each cryptographic key are as shown in [Table 6-3] below.

Classification	Cryptographic key	List of Standards	Cryptographic Key Generation Algorithm	Cryptographic Key Sizes
Authentication token encryption	Authentication token encryption key (token cryptographic key)	ISO/IEC 18031	HASH_DRBG(SHA256)	128 bits
	Authentication token HMAC key (token MAC key)	ISO/IEC 18031	HASH_DRBG(SHA256)	128 bits
	Authentication token cryptographic key encryption key (group key)	ISO/IEC 18031	HASH_DRBG(SHA256)	128 bits
TSF data	TSF data encryption key (TSF cryptographic key)	ISO/IEC 18031	HASH_DRBG(SHA256)	128 bits

encryption	TSF data HMAC key (TSF MAC key)	ISO/IEC 18031	HASH_DRBG(SHA256)	128 bits
	Secure channel session key (session key)	ISO/IEC 18031	HASH_DRBG(SHA256)	128 bits
	Secure channel session HMAC key (session MAC key)	ISO/IEC 18031	HASH_DRBG(SHA256)	128 bits
	SafeAgent private key	ISO/IEC 18033-2	RSAES(SHA-256)	2048 bits
	SafeAgent public key	ISO/IEC 18033-2	RSAES(SHA-256)	2048 bits
	PolicyServer private key	ISO/IEC 18033-2	RSAES(SHA-256)	2048 bits
	PolicyServer public key	ISO/IEC 18033-2	RSAES(SHA-256)	2048 bits
	private key encryption key(KEK)	ISO/IEC_9797-2	PBKDF2(HMAC-SHA-256)	128 bits

[Table 6-3] Cryptographic key generation algorithm

Classification	Description
Cryptographic module name	XecureCrypto v2.0.1.1
Validation number	CM-153-2024.5
Validated level	Security level 1
Developer	Hancom WITH Inc.
Validation date	May 8, 2020

[Table 6-4] Details on the validated cryptographic module

6.2.2. Cryptographic key distribution

Secure channel session key, secure channel MAC key and token group key are transmitted through a secure channel (RSAES-OAEP) formed after the mutual authentication through a specified digital signature [RSA_PSS] that meets [ISO/IEC 18033-2].

Cryptographic key	Timing of distribution / cryptographic algorithm
Secure channel session key (session key)	Encrypt with SafeAgent's public key in PolicyServer and distribute to SafeAgent Distribute a session key in each mutual authentication process
Secure channel HMAC key (session MAC key)	Encrypt with SafeAgent's public key in PolicyServer and distribute to SafeAgent Distribute a session MAC key in the mutual authentication process
Token cryptographic key encryption key (group key)	Encrypt with SafeAgent's public key in PolicyServer and distribute to SafeAgent Distribute a group key in the mutual authentication process

[Table 6-5] Timing of distribution and distribution method of "session key, group key encryption key"

6.2.3. Cryptographic key destruction

For an encryption key loaded onto the memory during key generation, distribution and operation, random bits are all overwritten with 0x00 after the valid period to destroy the cryptographic key.

Classification of encryption	Cryptographic key	When deleted	Cryptographic key destruction method
Authentication token encryption	Authentication token encryption key (token cryptographic key)	Immediately after use	Overwrite with 0x00
	Authentication token HMAC key (token MAC key)	Immediately after use	Overwrite with 0x00
	Authentication token cryptographic key encryption key (group key)	Immediately after use	Overwrite with 0x00
TSF data encryption	TSF data encryption key (TSF cryptographic key)	Immediately after use	Overwrite with 0x00
	TSF data HMAC key (TSF MAC key)	Immediately after use	Overwrite with 0x00
	Secure channel session key (session key)	Immediately after use	Overwrite with 0x00
	Secure channel session	Immediately after use	Overwrite with 0x00

	HMAC key (session MAC key)		
	SafeAgent private key	Immediately after use	Overwrite with 0x00
	SafeAgent public key	Immediately after use	Overwrite with 0x00
	PolicyServer private key	Immediately after use	Overwrite with 0x00
	PolicyServer public key	Immediately after use	Overwrite with 0x00
	private key encryption key(KEK)	Immediately after use	Overwrite with 0x00

[Table 6-6] Cryptographic key destruction method

6.2.4. Cryptographic operation

The TOE performs cryptographic operations for user authentication token encryption by using approved cryptographic algorithms of validated cryptographic module "XecureCrypto v2.0.1.1" whose security and implementation conformance have been validated by the KCMVP. The validated cryptographic module is operated in the approved mode of operation during cryptographic operations. CBC mode is used when performing the encryption using the block cipher algorithm, and methods specified in TTAS.KO-12.004/R1 are applied in using IV.
Standards, cryptographic algorithm and cryptographic key size used for cryptographic operation during the user data encryption are listed in [Table 6-7] below.

Cryptographic operation	Standard	Cryptographic algorithm	Cryptographic key size
Block cipher	TTAS.KO-12.004/R1	SEED	128 bits
Message authentication	ISO/IEC_9797-2	HMAC-SHA-256	128 bits
Hash	ISO/IEC_10118-3	SHA256	-

[Table 6-7] Cryptographic operation algorithm (authentication token encryption)

The TOE performs cryptographic operations for TSF data encryption by using approved cryptographic algorithms of validated cryptographic module "XecureCrypto v2.0.1.1" whose security and implementation conformance have been validated by the KCMVP. The validated cryptographic module is operated in the approved mode of operation during cryptographic operations. CBC mode is used when performing the encryption using the block cipher algorithm, and methods specified in KS X 1213 and TTAS.KO-12.004/R1 are applied in using

IV. Standards, cryptographic algorithm and cryptographic key size used for cryptographic operation during the TSF data encryption are listed in [Table 6-8] below.

Cryptographic operation	Standard	Cryptographic algorithm	Cryptographic key size
Block cipher	TTAS.KO-12.004/R1	SEED	128 bits
	KS X 1213	ARIA	128 bits

[Table 6-8] Cryptographic operation algorithm (TSF data encryption)

The TOE performs cryptographic operations for TOE internal mutual authentication and key exchange by using approved cryptographic algorithms of validated cryptographic module "XecureCrypto v2.0.1.1" whose security and implementation conformance have been validated by the KCMVP. The validated cryptographic module is operated in the approved mode of operation during cryptographic operations. During the mutual authentication, RSA digital signature is made with a private key, and the digital signature is verified with a public key for mutual verification. In this case, RSA-PSS digital signature algorithm used along with SHA-256 is an approved algorithm. For key exchange, RSA public key encryption is used for key exchange. In this case, RSAES-OAEP algorithm used along with SHA-256 is an approved algorithm. Standards, cryptographic algorithm and cryptographic key size used for cryptographic operation during the mutual authentication and key exchange are listed in [Table 6-9] below.

.

Cryptographic operation	Standard	Cryptographic algorithm	Cryptographic key size
Key encryption (key exchange)	ISO/IEC 18033-2	RSAES-OAEP (SHA-256)	2048 bits
Mutual authentication	ISO/IEC 14888-2	RSA-PSS (SHA-256)	2048 bits

[Table 6-9] Cryptographic operation algorithm (mutual authentication and key exchange encryption)

The TOE performs cryptographic operations for encrypted communication between TOE components by using approved cryptographic algorithms of validated cryptographic module "XecureCrypto v2.0.1.1" whose security and implementation conformance have been validated by the KCMVP. The validated cryptographic module is operated in the approved mode of operation during cryptographic operations. When using block cipher algorithms to perform

the encryption, ECB mode is not used regardless of a plaintext size. Methods specified in TTAS.KO-12.004/R1 are applied to the use of IV in CBC, CFB and OFB modes, and to the use of counter in IV and CTR modes.
Standards, cryptographic algorithm and cryptographic key size used for encrypted communication between TOE components are listed in [Table 6-10] below.

Cryptographic operation	Standard	Cryptographic algorithm	Cryptographic key size
Encrypted communication	TTAS.KO-12.004/R1	SEED	128 bits
Message authentication	ISO/IEC_9797-2	HMAC-SHA-256	128 bits

[Table 6-10] Cryptographic operation algorithm (encrypted communication)

The TOE performs cryptographic operations for integrity verification by using approved cryptographic algorithms of validated cryptographic module "XecureCrypto v2.0.1.1" whose security and implementation conformance have been validated by the KCMVP. The validated cryptographic module is operated in the approved mode of operation during cryptographic operations. The integrity is verified through hash-based verification, using SHA256 algorithm, an approved algorithm. Hash values for integrity verification are encrypted by using block cipher algorithms, and the integrity of hash values is protected by using HMAC values. Standards, cryptographic algorithm and cryptographic key size used for integrity verification are listed in [Table 6-11] below.

Cryptographic operation	Standard	Cryptographic algorithm	Cryptographic key size
Block cipher	TTAS.KO-12.004/R1	SEED	128 bits
Integrity verification of hash data	ISO/IEC_9797-2	HMAC-SHA-256	128 bits
Hash verification	ISO/IEC_10118-3	SHA256	-

[Table 6-11] Cryptographic operation algorithm (integrity verification)

6.2.5. Random bit generation

The TOE generates random bits necessary for cryptographic key generation by using HASH_DRBG (SHA256) algorithm with the random bit generator of "XecureCrypto v2.0.1.1, a validated cryptographic module whose security and implementation conformance have been validated by the KCMVP.
Information on the validated cryptographic module is described in [Table 6-4].

※ Relevant SFRs

- FCS_CKM.1, FCS_CKM.2, FCS_CKM.4, FCS_COP.1, FCS_RBG.1

6.3. Identification and Authentication

The TOE satisfies the initial identification and authentication of end users and the identification and authentication of the authorized administrator by verifying passwords. In addition, the TOE verifies authentication tokens after the initial identification and authentication of end users to satisfy the identification and authentication of end users.

6.3.1. Authentication failure handling

If the identification and authentication of an end user to access a business system fails for 5 consecutive times, an attempt by the user account to access the business system is blocked for 10 minutes, and audit records on the authentication failure are stored. Also, if the identification and authentication of the security administrator for the management access fails for 5 consecutive times, an attempt by the administrator account for the management access is blocked for 10 minutes, and audit records on the authentication failure are stored.

6.3.2. TOE internal mutual authentication

Mutual authentication between SafeAgent and PolicyServer is performed with an authentication protocol developed by Hancom WITH Inc.

Upon the start-up of the TOE and when calling the initial authentication function and the token authentication function, TOE components of SafeAgent and PolicyServer respectively receive message signature values encrypted with a private key of the other TOE component and verifies them, thereby performing the mutual authentication. Additionally, session keys and session MAC keys are shared for encrypted communication of a secure channel between SafeAgent and PolicyServer.

Classification	Standard	Cryptographic algorithm	Cryptographic key size
Signature verification	ISO/IEC 14888-2	RSA-PSS (SHA-256)	2048 bits
Encryption	ISO/IEC 18033-2	RSAES-OAEP (SHA-256)	2048 bits

[Table 6-12] Cryptographic operation algorithm for mutual authentication

6.3.3. Verification of secrets

A password of an end user using SSO shall be at least 9 up to 64 digits containing a combination of all three types of English alphabet, number and special character. A password of the security administrator shall be at least 9 up to 64 digits containing a combination of all three types of English alphabet, number and special character (!@#$%^&*()_+-=).
For SSO, it is required to use authentication tokens generated by the TOE in accordance with the defined criteria. Authentication tokens are destroyed by means of zeroization of authentication token values and TokenID values.

6.3.4. Generation of secrets

An authentication token is generated and verified to process the user identification and authentication for SSO.

In an authentication token, user information and authentication information are encrypted with SEED-CBC, and the integrity of the token is protected with HMAC-SHA-256.

Classification	Standard	Cryptographic algorithm	Cryptographic key size
Block cipher	TTAS.KO-12.004/R1	SEED	128 bits
Message authentication	ISO/IEC_9797-2	HMAC-SHA-256	128 bits

[Table 6-13] Cryptographic operation algorithm for authentication token

6.3.5. Destruction of secrets

If a token expiry function is called for an authentication token used for SSO, authentication information is destroyed through zeroization to zeroize TokenID value of the user authentication token stored on the PolicyServer memory. If the process to issue or verify an authentication token in SafeAgent is completed, the authentication token value stored on the memory is zeroized to "0" to destroy authentication information.

6.3.6. Authentication / Identification

The TOE satisfies an end user's initial identification and authentication, and the authorized administrator's identification and authentication through the password verification.
In addition, the TOE satisfies an end user's identification and authentication by verifying an authentication token after satisfying the end user's initial identification and authentication.

6.3.7. Single-use authentication mechanism

The TOE use a session ID prevent the reuse of authentication data, and use TokenID issued by PolicyServer to prevent the reuse of authentication tokens.
After the token-based authentication is performed, a new TokenID is assigned for the issued authentication token to renew the authentication token. The TOE provide a function to confirm the validity of the authentication with a newly issued authentication token in order to ensure the reuse of authentication information.

6.3.8. Protected authentication feedback

The identification and authentication shall be performed before allowing access to and control of any security functions of the TOE. If the authentication fails, the TOE does not provide feedback on the reason for the failure. A password entered while the authentication is in progress or a password is changed is masked with "*" to prevent it from being disclosed.

※ Relevant SFRs

- FIA_AFL.1, FIA_IMA.1, FIA_SOS.1, FIA_SOS.2, FIA_SOS.3, FIA_UAU.1, FIA_UAU.4, FIA_UAU.7, FIA_UID.1

6.4. Security Management

The security administrator shall perform the security management function through the security management interface (web browser), and can use the security management function only after going through the identification and authentication process. The security management function is classified into the management of security functions behaviors, the management of TSF data, and the management of ID and password.

6.4.1. Management of security functions behavior

In the TOE, the security management access and control function is called only when the identification and authentication process, which is enforced by the TOE, is successfully performed. The authorized administrator (security administrator) is allowed to access the security management interface through a secure channel (SSL) which is an operational environment.
Provides the security administrator with the function to enable and disable security functions of end users access policy.

6.4.2. Management of TSF data

The TOE provides the management function to enable the authorized administrator (top administrator) to modify, query, delete and add (new) TSF data listed in [Table 6-14].

Security function	Management behavior
Administrator information management	Query
	Modify
End user management	Query
	New(Add)
	Modify
	Delete
Organization management	Query
	New(Add)
	Modify
	Delete

[Table 6-14] List of TSF data management

6.4.3. Management of ID and password (extended)

The TOE enforces the administrator to change his/her password when accessing the security management interface for the first time. A password combination rule is as follows:
- The combination rule to contain at least 3 types of letters among English alphabets, numbers and special characters(!, @, #, $, %, ^, &, *, (,), _, +, -, =), in at least 9 up to 64 digits

6.4.4. Security roles

For the security roles related to the security, there is a single administrator role without separate roles for individual administrator. Security functions such as managing TSF data, user accounts, user profiles and passwords, administrator alarm emails, security management access IPs, etc. are restricted to the authorized administrator.

※ Relevant SFRs

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

6.5. Protection of the TSF

6.5.1. Basic internal TSF data transfer protection

When TSF data are transmitted between separate parts of the TOE, the transmitted TSF data such as audit data and critical security parameters are protected against disclosure and modification.

For secure communication between SafeAgent and PolicyServer, a secure channel communication is provided by using a session key and a session MAC key exchanged during the mutual authentication. The transmitted TSF data are encrypted with SEED-CBC cryptographic operation, and the integrity of the transmitted TSF data is protected with HMAC-SHA-256.

Classification	TSF data	List of standards	Cryptographic algorithm	Cryptographic key size
Encrypted communication	Transmitted TSF data	TTAS.KO-12.004/R1	SEED	128 bits
		ISO/IEC_9797-2	HMAC-SHA-256	128 bits

[Table 6-15] Cryptographic algorithm for internal TSF data transfer

6.5.2. Basic protection of stored TSF data (extended)

The TOE protects the security administrator and end user passwords, authentication token information, cryptographic keys, critical security parameters, TOE set values (security policy, configuration parameter), audit data and so forth from unauthorized disclosure and modification. Especially, TSF data such as the security administrator and end user passwords, authentication tokens, critical security parameters, TOE set values and DBMS connection information are encrypted with approved cryptographic algorithms of validated cryptographic module "XecureCrypto v2.0.1.1," and then stored.

Private keys of PolicyServer and SafeAgent, which are KEKs, are encrypted with ARIA128 algorithm through key derivation with PBES2 (PBKDF2).

6.5.3. TSF self test

The TOE provides a function to run self tests of critical processes and the validated cryptographic module during the initial start-up and in 10-minute intervals during normal operation to ensure that an abnormal condition (for example, error, shutdown, etc.) of SafeAgent or PolicyServer, which are TOE components, does not affect major functions and

security functions. In the event of an abnormal condition, it is notified to the authorized administrator via email.
The TOE verifies the integrity of all configuration files, executable files and TSF data necessary for the operation of the TOE. The integrity verification is carried out during the initial start-up and in 10-minute intervals during normal operation. In addition, if the integrity verification finds any violation, it is notified to the authorized administrator via email.

※ Relevant SFRs

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

6.6. TOE Access

6.6.1. Per user attribute limitation on multiple concurrent sessions

The TOE restricts the maximum number of concurrent sessions that belong to the same user to 1 in order to block concurrent access to the same account. Since there is only one account for the sole top-level security administrator of the TOE, concurrent access to the same privilege is also not possible.

6.6.2. Management of TSF-initiated sessions (extended)

The TOE terminates a session if an end user or the authorized administrator does not access SSO or the administrator interface, or stays inactive for 10 minutes after the login.

6.6.3. TOE session establishment

The TOE controls TOE access by allowing access to the security management interface only from the registered IPs (2 or less by default). An accessible IP can be set during the TOE is installed. It is also possible to add, modify and delete accessible IPs after the installation by setting a list of IPs allowed to log in to the security management. When setting IPs, it is not allowed to add an IP by designating an IP address range, but needs to add each IP address individually. Settings such as 0.0.0.0 and 255.255.255.255, which means the entire range of the network, are not allowed.

※ Relevant SFRs

- FTA_MCS.2, FTA_SSL.5, FTA_TSE.1