ISign+ v3.0

Security Target Lite

D-ST : 1.0

Penta SECURITY

Revision History

Version	Description	Author	Release Date
1.0	Sanitized version of the ST D-ST : 1.4	Eunyoung Kim	2019.03.11

Table of Contents

Figures

Tables

1 ST Introduction

1.1 ST Reference

Table 1-1 ST Reference

Title	ISign+ v3.0 Security Target Lite
Version	v1.0
Publication Date	2019-03-11
Author	Penta Security System Inc.
Common Criteria version	CC V3.1 R5
Protection Profile	Korean National Protection Profile for Single Sign On V1.0
Evaluation Assurance Level	EAL1+(ATE_FUN.1)
Keywords	Single Sign On, SSO

1.2 TOE Reference

Table 1-2 TOE Reference

Item	Specification		Release Method
TOE	ISign+ v3.0		-
Version	v3.0.27		-
Components	SSO Server	• SS-ATH v3.0.27 • ISign+_v3.0_SS-ATH_v3.0.27.tar	CD
	SSO Agent	• SA-WEB v3.0.27 • ISign+_v3.0_SA-WEB_v3.0.27.tar	CD
	Preparative Procedure	• ISign+ v3.0 Preparative Procedures U-IG : 1.8 • UIG_ISign+_v3.0_Preparative Procedures_v1.8.pdf	CD
	Operation Guide	• ISign+ v3.0 Operational Guide U-OG : 1.5 • UOG_ISign+_v3.0_Operational Guide_v1.5.pdf	CD
Developer	Penta Security System Inc.		-

1.3 TOE Overview

1.3.1 TOE usage

ISign+ v3.0 (hereinafter referred to as “TOE”) performs end-user identification and authentication to enable the user to access various business systems and use the service through a single user login without additional login action.

The TOE performs end-user identification and authentication based on the ID / PW information of the end-user pre-registered with the SSO server. The token-based authentication is performed when an authentication token is issued through the initial authentication and then the user wants to login to another business system.

The procedure of end-user identification and authentication of TOE is as <Figure 1-1 End-user identification and authentication procedure>.

The end-user identification and authentication procedure can be grouped into the initial authentication phase using the ID/PW, and the token-based authentication phase that accesses the business system using the token issued during the initial authentication procedure.

The execution procedure of the initial authentication phase is as follows. (1) When the user accesses the business server, the SSO agent redirects to the SSO server. Then, the user enters ID / PW in the login page of the SSO server to request a user login. (2) SSO server performs login verification using user information stored in DBMS. The SSO server issues an authentication token if the login validation result is valid. (3) The user requests authentication token verification from the SSO agent. (4) The SSO agent requests authentication token verification from the SSO server. (5) The SSO server validates the authentication token and returns the result. At this time, the authentication token is updated. The SSO agent allows the user to use the business service if the authentication token verification succeeds (Business service login success).

The token-based authentication step is performed only when the authentication token is successfully issued through the initial authentication step. (6) When the user accesses the business server, the SSO agent redirects to the SSO server. Since the SSO server already has an SSO session for the user, it extracts the authentication token from the SSO session and sends it to the user. The user sends an authentication token verification request to the SSO agent. (7) The SSO agent requests authentication token verification from the SSO server. (8) The SSO server validates the authentication token and returns the result. At this time, the authentication token is updated. The SSO agent allows the user to use the business service if the authentication token verification succeeds (Business service login success).

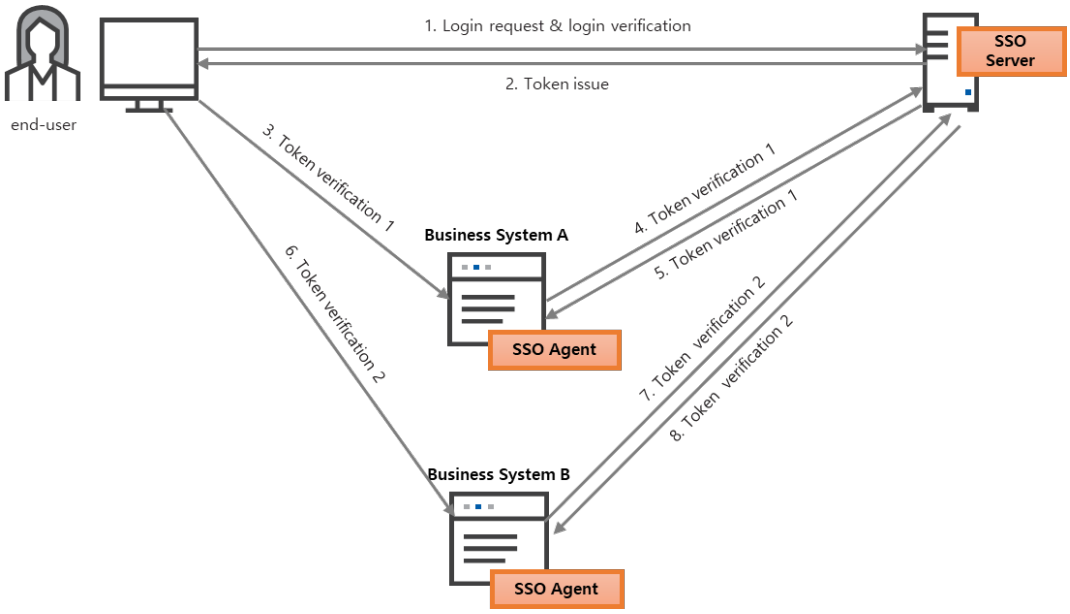


Figure 1-1 End-user identification and authentication procedure

1.3.2 TOE major security features

The major security features of the TOE are as follows.

- Security audit

The TOE generates and stores audit data in case of major security events where the management console administrator and the end-user use the TSF. And it provides the management console

administrator with an audit review function that can inquire audit data. It also detects potential security violation events to perform appropriate actions. And it performs audit data loss prediction and prevention functions.

- Cryptographic support

  The TOE generates and distributes cryptographic keys used for cryptographic operations required by the TOE using CIS-CC v3.3, a validated cryptographic module, performs cryptographic operations, and destroys the plaintext cryptographic keys loaded in memory.

- Identification and authentication

  The TOE provides ID / PW-based identification and authentication functions to the end-user and the management console administrator. It protects authentication feedback during the authentication process. In case of continuous authentication failure, TOE performs authentication failure handling according to the setting of management console.

  The TOE provides identification and authentication function based on authentication token for the end-user and performs mutual authentication between SSO server and SSO agent.

  The TOE blocks attempt to reuse ID/PW authentication information for the end-user and the administrator, and it also performs the reuse prevention function of the end-user authentication token.

  The TOE safely destroys the authentication tokens (plain text and ciphertext) loaded into memory after they are used.

- Security management

  The TOE provides management console administrators with management functions of TSF, TSF data and the password of management console administrator and end-user.

- TSF protection

  The TOE protects TSF data transmitted between SSO server and SSO agent by using CIS-CC v3.3, which is a validated cryptographic module. The critical TSF data is encrypted and protected with the validated cryptographic module. Also, the TOE verifies the availability of the main external entities (mail server, DMBS, and WAS) and performs TSF self-test and integrity check of TSF / TSF data.

- TOE access

  The SSO server allows only from the terminal with designated IP address for the management console administrator's management access. And it limits the number of concurrent sessions. Also, if the administrator or the end-user session exceeds the set inactivity time, the session is terminated.

1.3.3 TOE type

The TOE defined in this Security Target is a 'Single Sign On' system that enables the end-user to access various business systems through a single user login, consists of SSO agent and SSO server. The SSO server performs user login processing, authentication token issuance and management, and authentication token validation. The SSO agent is installed in each business system, requests authentication token verification, and transmits the authentication token verification result to the business service.

This ST defines the security functional requirements provided by the SSO agent and the SSO server, which are mandatory TOE components, and the TOE shall implement these security functional requirements.

1.3.4 TOE Operational Environment

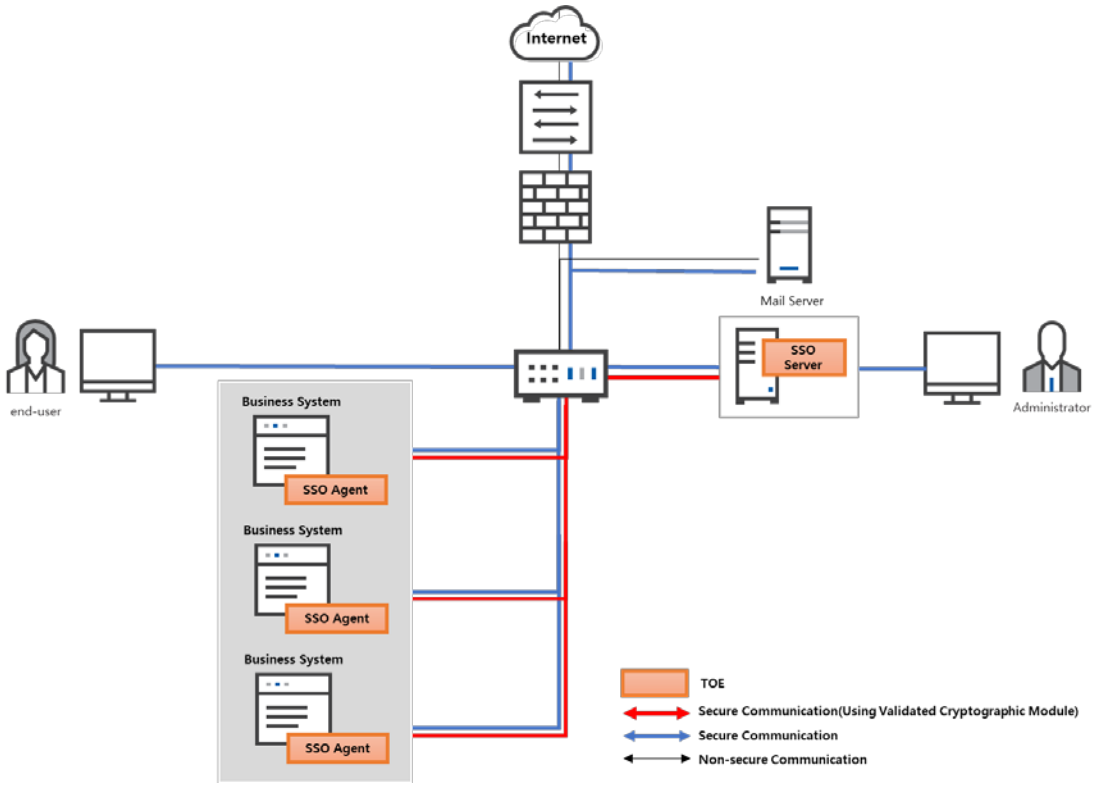


Figure 1-2 TOE Operating Environment

The TOE operating environment is as <Figure 1-2 TOE Operating Environment>> shows. When the end-user or the management console administrator accesses TOE through web browser, WAS, which is operating environment of SSO agent and SSO server supports HTTPS TLS based communication.

The TOE consists of SSO server and SSO agent. Using user information stored in the DBMS, the SSO server provides various functions such as direct user login verification, authentication token issuance and management/policy setting. The SSO agent also provides various functions such as the authentication token verification request, and is installed and operated on each system. The external IT entities required to operate the TOE are as shown in <Table 1-3 External IT entities required for TOE>. The following external IT entities are not included in the TOE scope, but their interfaces communicating with the TOE are included in the evaluation area.

Table 1-3 External IT entities required for TOE operation

External IT entity	Description
Mail server	Mail server to send mails, such as management console administrator notification when audit data loss is predicted
Admin PC	PC used by administrator to connect to the management console
User PC	PC used by end-user to use SSO of the TOE

The operating environment for each component of the TOE is as follows.

Table 1-4 SW Operating environment of the SSO server

Type	SW name & version	Description and requirement
OS	Debian GNU/Linux 8.9(jessie) (kernel 3.16.59-1) 64bits	• As a Linux based OS, provides reliable time information
DBMS	MariaDB v10.2.22 64bits	• This DBMS is used to securely store the TOE audit data and TSF data

WAS	Apache Tomcat v8.5.35 (openjdk 1.8.0_202) 64bits	• Web Application Server required to operate SSO server core logic and web-based management console • Supports TLS v1.2 based HTTPS communication

Table 1-5 HW Requirements for SSO Server

Type	Specification
CPU	Intel Pentium Processor G4600 3M Cache 3.60 GHz or higher
Memory	8 GB or higher
HDD	500 MB or higher (space for TOE installation)
NIC	100/1000 Mbps x 1EA or higher

Table 1-6 SW Requirements for SSO agent

Type	SW name & version	Description and requirements
OS	Debian GNU/Linux 8.9(jessie) (kernel 3.16.59-1) 64bits	• As a Linux based OS, provides reliable time information
WAS	Apache Tomcat v8.5.35 (openjdk 1.8.0_202) 64bits	• Web Application Server required to operate web-based business service and the SSO agent • Supports TLS v1.2 based HTTPS communication

Table 1-7 HW requirements for the SSO agent

Type	Specification
CPU	Intel Pentium Processor G4600 3M Cache 3.60 GHz or higher
Memory	8 GB or higher
HDD	10 MB or higher (space for TOE installation)
NIC	100/1000 Mbps x 1EA or higher

Table 1-8 SW Requirements for the management console administrator and user's PC

Type	SW name version
OS	Windows 10 Pro (64-bit)
SW	Chrome 71.0.3578.98(official build) (64-bit)

Table 1-9 HW Requirements for the management console administrator and user's PC

Type	Specification
CPU	Intel core i5-4200U 1.60 GHz or higher
Memory	4 GB or higher
HDD	100 GB or higher
NIC	100/1000 Mbps x 1EA or higher

1.4 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**.

1.5 TOE description

1.5.1 Physical scope of the TOE

The TOE consists of SSO Agent, SSO Server, Preparative Procedure and Operation Guide.

The SSO server is a set of software that consists of an authentication server, that manages initial user authentication and authentication-token-based SSO, and several script files and management console. This SSO server is delivered via CD. TOE installation and operation requirements such as hardware, operating system, WAS, and DBMS are not included in the TOE.

The SSO agent is the entity that requests the SSO server to verify the authentication token that is submitted by the end user's PC. The SSO agent is installed in a Web-based business system developed in JSP and forms a part of the business system to provide SSO functionality. The SSO agent is delivered via CD, and it requires additional development to apply to business systems, depending on each business system environment.

Table 1-10 Physical scope of the TOE

Type	File name	Format	Delivery Method
SSO server	ISign+_v3.0_SS-ATH_v3.0.27.tar	SW	CD
SSO agent	ISign+_v3.0_SA-WEB_v3.0.27.tar	SW	CD
Preparative Procedure	UIG_ISign+_v3.0_Preparative Procedures_v1.8.pdf	pdf file	CD
Operation Guide	UOG_ISign+_v3.0_Operational Guide_v1.5.pdf	pdf file	CD

Validated cryptographic modules installed in the TOE are as follows:

Table 1-11 Validated cryptographic modules

cryptographic module & version	Validation number	Validation date	Developer
CIS-CC v3.3	CM-145-2023.11	2018-11-07	Penta Security System

Validated cryptographic modules are respectively distributed with the SSO server and SSO agent.

1.5.2 Logical scope of the TOE

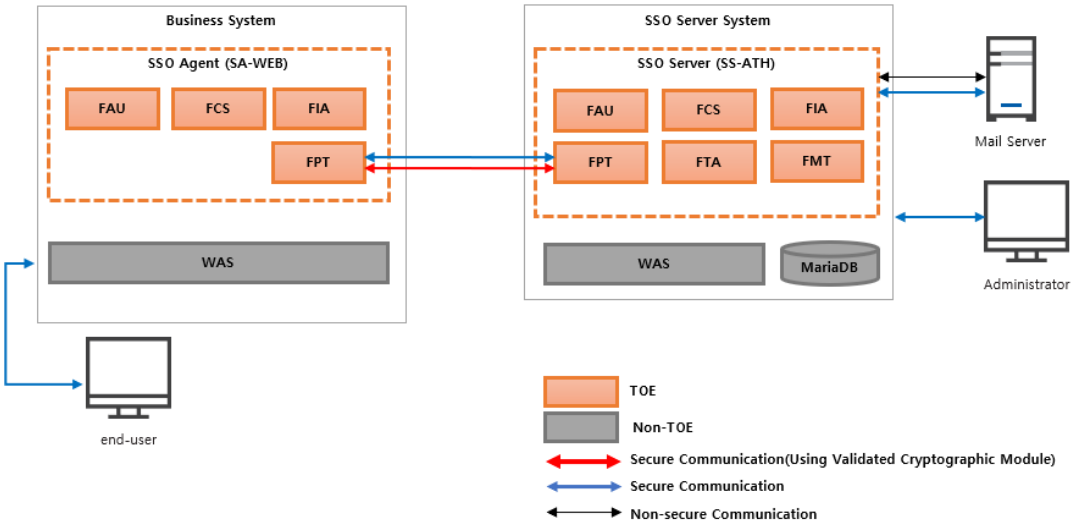


Figure 1-3 Logical scope of the TOE

The SSO server and the SSO agent conduct the following security functions:

- Security audit (FAU)

  The SSO server generates audit data for security audit events that occur on SSO server and stores / manages audit data in the DBMS. The SSO server provides the normal/selectable audit review function for the audit data generated by the SSO agent as well as the SSO server audit data. In addition, the SSO server conducts appropriate security alert functions through the analysis of potential audit data violations. The TSF shall notify the 0-level administrator of possible loss of audit data by email. And if the audit storage is full, the TSF performs an overwrite of the oldest stored audit record and an email notification to the 0-level administrator.

- Encryption support (FCS)

  The following encryption support is conducted to execute mutual authentication between the SSO server and the SSO agent, to protect the transmitted TSF data, to generate the authentication token, to protect the stored TSF data, and to authenticate the integrity of the TSF and TSF data.

  - Cryptographic key generation: KEK, DEK, Integrity-verification key, Token en/decryption key
  - Cryptographic key destruction: destroys the plain cryptographic key loaded into memory (zeroize)
  - Cryptographic key distribution: from the SSO server, download salt.dat file containing DEK, integrity verification key, administrator information, and the salt value required to derive the same KEK between the SSO server and the SSO agent. Then, store it in USB memory and distribute it offline to the SSO agent.
  - Cryptographic operation: executes the cryptographic operations described in <Table 5-6 Cryptographic operation (symmetric key)>, TSF and TSF data integrity authentication, password encryption for management console administrator and end-user.

- Identification and authentication (FIA)

  Identification and authentication are conducted for the management console administrator who manages TOE security and end-users (ID/PW-based initial authentication, authentication-token-based

authentication) who use SSO. The TOE prevents reuse and provides authentication feedback protection for the administrator and end-user's authentication data. If authentication fails continuously, the authentication functions will be deactivated for the configured time period. If the end-user's initial authentication is successful, a token for authentication-token-based authentication is generated and issued. When an authentication token is created and authenticated, it is initialized and deleted in the memory.

- TOE access (FTA)

  The number of concurrent sessions is limited by the level of authority of the management console administrator (level 0~2), and level 3 administrator can use an unlimited number of sessions. The session ends when the management console administrator is inactive for a specified amount of time. Management console administrators can access management connection session only through the allowed IPs.

- TSF protection (FPT)

  The SSO server performs self-tests on the authentication token generation and verification function. The SSO server and the SSO agent also conduct integrity inspection on the TSF and TSF data. While the SSO server communicates with the SSO agent, encryption is conducted using a validated cryptographic module to protect the TSF data. The operating environment, MariaDB, stores TSF data such as various cryptographic keys and authentication data. In order to protect the TSF data to be stored in MariaDB, important TSF data is encrypted using the validated cryptographic module. Tests for external entities such as mail servers, DBMSs, and WAS will be conducted for availability as well.

- Security management (FMT)

  The TOE provides the management console administrator with security function management, TF data management, and password combination rule management functions. 0~3-level administrators use the management console for each role to manage TSF data, security functions, and password combination rules. The role of the administrator can be found in Section 1.6 (Terms and definitions).

1.6 Terms and definitions

Most terms used in this ST are consist with the Common Criteria for Information Technology Security Evaluation. Additional terms used only in this ST are as follows.

- Management Console

  A Web-based user interface consisting of several web pages for audit review and management of TSF and TSF data of the SSO server.

- Management console administrator / Authorized administrator

  Authorized user to securely operate and manage the TOE. There are 0~3-level administrators based on their access levels.

- 0-level administrator

  Management console administrator created initially when the SSO server installed. Its ID is 'adm' which is fixed and not changeable. 0-level admin can operate all the functions of offered by the TOE. And only the administrator can use the [Integrity Verification] function and reboot the SSO server and the SSO agent.

- 1-level administrator

  Management console administrators who can use Inquiry/Management/Setting function of SSO Authentication Server, all the functions of Integrated Management and Product Registration function.

- 2-level administrator

Management console administrators who can view management logs and user logs, manage accounts of users and download the manual.

- 3-level administrator

  Management console administrators can view management logs and user logs, download the manual.

- end-user

  Users of the TOE who want to use the business system, not the administrators of the TOE

- agent ID

  A Unique ID assigned automatically when a business service information added on the management console.

- Token serial number

  One of the means for preventing reuse of the authentication token is to generate a token serial number for each user session. When an authentication token is issued, the first '1' is given, and when the authentication token is reissued (updated) after the authentication token verification, the SSO server increases by one.

2 Conformance claim

2.1 CC conformance claim

Table 2-1 CC conformance claim

Item	Description
Common Criteria (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)
CC Part2: Security Functional Components	Extended: FCS_RBG.1, FIA_IMA.1, FIA_SOS.3, FMT_PWD.1, FPT_PST.1, FTA_SSL.5
CC Part3: Security Assurance Components	Conformant
Package	Augmented: EAL1 augmented (ATE_FUN.1)

2.2 PP conformance claim

This ST claim conformance the following PP.

- Korean National Protection Profile for Single Sign On V1.0

2.3 Package conformance claim

This ST claims conformance to assurance package EAL1 augmented with ATE_FUN.1.

2.4 Conformance claim rationale

This ST claims conformance to security objectives and security requirements by “strict PP conformance” adherence to ‘Korean National Protection Profile for Single Sign On V1.0’.

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.

3.1 Security objectives for the operational environment

Item	Security objective
OE.PHYSICAL_CONTROL	The place where SSO agent and SSO server among the TOE components are installed and operated shall be equipped with access control and protection facilities so that only authorized administrator can access.
OE.TRUSTED_ADMIN	The authorized administrator of the TOE shall be non-malicious users, have appropriately trained for the TOE management functions and accurately fulfill the duties in accordance with administrator guidances.
OE.LOG_BACKUP	The authorized administrator shall periodically check a spare space of audit data storage in case of the audit data loss, and carries out the audit data backup (external log server or separate storage device, etc.) to prevent audit data loss.
OE.OPERATION_SYSTEM_REINFORCEMENT	The authorized administrator of the TOE shall ensure the reliability and security of the operating system by performing the reinforcement on the latest vulnerabilities of the operating system in which the TOE is installed and operated.
OE.SECURE_DEVELOPMENT	The 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 of the manual provided with the TOE.
OE.TIMESTAMP	The TOE operational environment shall provide reliable time stamps to the TOE.
OE.DBMS	DBMS that saves the TSF data and audit data is operated in a physically safe environment.
OE.TRUSTED_CHANNEL	The TOE operational environment shall provide a trusted channel between the SSO server and the SSO agent to protect user data other than TSF data.

4 Extended components definition

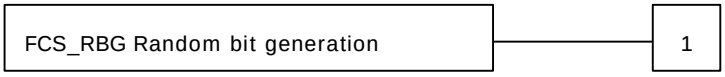
4.1 Cryptographic support

4.1.1 Random Bit Generation

- Family Behaviour

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

- Component leveling



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

- Management: FCS_RBG.1

  There are no management activities foreseen.

- Audit: FCS_RBG.1

  There are no auditable events foreseen.

4.1.1.1 FCS_RBG.1 Random bit generation

- Hierarchical to: No other components.
- Dependencies: No dependencies.
- FCS_RBG.1.1

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

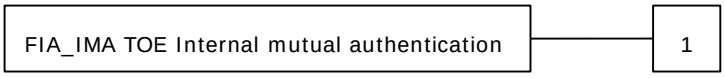
4.2 Identification & authentication

4.2.1 TOE Internal mutual authentication

- Family Behaviour

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

- Component leveling



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

- Management: FIA_IMA.1

  There are no management activities foreseen.

- Audit: FIA_IMA.1

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

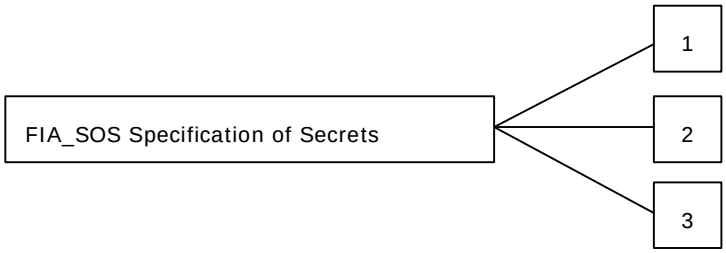
  The TSF shall perform mutual authentication between [assignment: *different parts of TOE*] using the [assignment: *authentication protocol*] that meets the following [assignment: *list of standards*].

4.2.2 Specification of Secrets

- Family Behaviour

  This family defines requirements for mechanisms that enforce defined quality metrics on provided secrets 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 now composed 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 to the 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*].

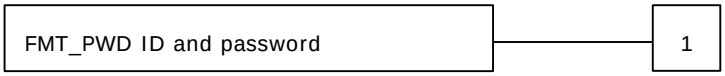
4.3 Security Management

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

- Management: FMT_PWD.1

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

  a) Management of ID and password configuration rules.

- Audit: FMT_PWD.1

  The following actions are recommended to record if FAU_GEN Security audit data generation is included in 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 special characters unusable for password, etc.*]

- FMT_PWD.1.2

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

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

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

- FMT_PWD.1.3

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

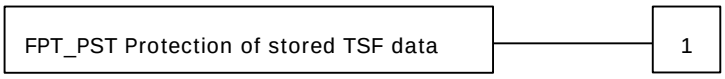
4.4 Protection of the TSF

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 from the unauthorized modification or disclosure.

- Component leveling



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

- Management: FPT_PST.1

  There are no management activities foreseen.

- Audit: FPT_PST.1

  There are no auditable events foreseen.

4.4.1.1 FPT_PST.1 Basic protection of stored TSF data

- Hierarchical to: No other components.
- Dependencies: No dependencies.
- FPT_PST.1.1

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

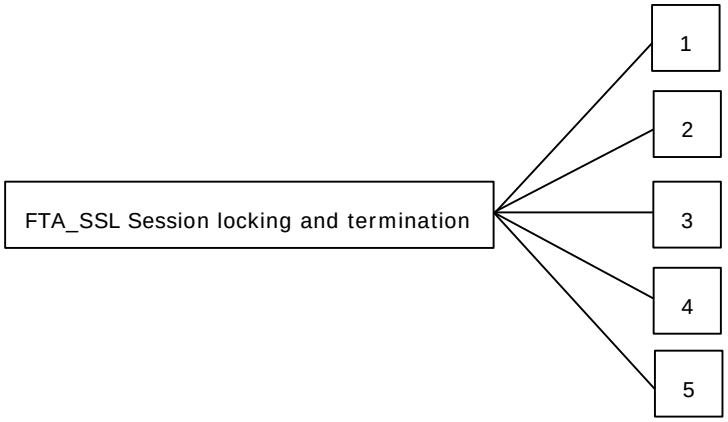
4.5 TOE Access

4.5.1 Session locking and termination

- Family Behaviour

  This family defines requirements for the TSF to provide the capability for TSF-initiated and user-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 or terminates the session after a specified time interval of user inactivity.

■ Management: FTA_SSL.5

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

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

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

■ Audit: FTA_SSL.5

The following actions are recommended to record if FAU_GEN Security audit data generation is included 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 the session*,

• *terminate*] an interactive session after a [assignment: *time interval of user inactivity*].

5 Security requirements

The security requirements specify security functional requirements and assurance requirements that must be satisfied.

The security requirements of this ST conform to the security requirements of "Korean National PP for Single Sign On V1.0 ".

5.1 Security functional requirements

In "Korean National PP for Single Sign On V1.0" the security functional requirements are classified into mandatory SFRs and optional SFRs. The SFRs defined in this Security Target comprise all the mandatory SFRs and some optional SFRs of the PP.

The following [Table 5-1 Summary of Security Functional Requirements] summarizes the security functional requirements defined in this Security Target.

Table 5-1 Summary of Security Functional Requirements

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(1)	Selectable audit review (end-use log)
	FAU_SAR.3(2)	Selectable audit review (administrator log)
	FAU_STG.3	Action in case of possible audit data loss
	FAU_STG.4	Prevention of audit data loss
FCS	FCS_CKM.1(1)	Cryptographic key generation (KEK generation)
	FCS_CKM.1(2)	Cryptographic key generation (Private keys other than KEK)
	FCS_CKM.2	Cryptographic key distribution
	FCS_CKM.4	Cryptographic key destruction
	FCS_COP.1(1)	Cryptographic operation (symmetric key)
	FCS_COP.1(2)	Cryptographic operation (HMAC)
	FCS_COP.1(3)	Cryptographic operation (HASH)
	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
	FIA_UAU.2(1)	User authentication before any action (end-user)
	FIA_UAU.2(2)	User authentication before any action (administrator)
	FIA_UAU.4	Single-use authentication mechanisms
	FIA_UAU.7	Protected authentication feedback
	FIA_UID.2(1)	User identification before any action (end-user)

	FIA_UID.2(2)	User identification before any action (administrator)
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_TEE.1	Testing of external entities
	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

5.1.1 Security audit (FAU)

5.1.1.1 FAU_ARP.1 Security alarms

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

FAU_ARP.1.1

The TSF shall take [actions on <Table 5-2 Actions for potential security violation>] upon detection of a potential security violation.

Table 5-2 Actions for potential security violation

Security functional component	Potential security violation	Actions
FAU_STG.3	The audit trail storage is checked (every 1 minute) and if the audit trail exceeds the storage capacity threshold	1) sending an e-mail to the 0-level administrator
FAU_STG.4	The audit trail storage is checked (every 1 minute) and if the audit trail exceeds the storage capacity limit	1) sending an e-mail to the 0-level administrator 2) Deletes from the DB partition where the oldest audit records are stored until it becomes less than the threshold value set in FAU_STG.3
FIA_AFL.1 FIA_UAU.2(1)(2)	When the defined number of unsuccessful authentication attempts of an administrator or an end-user has been met	Disable authentication for the administrator and the end-user for the amount of time 0~1-level administrators can define
FPT_TST.1	Authentication token generation and verification self-test failed	1) sending an e-mail to the 0-level administrator 2) afterwards, processing authentication request failure for all end-users
FPT_TST.1	TSF integrity verification failure	The SSO server performs as follows: 1) sending an e-mail to the 0-level administrator The SSO agent performs as follows:

		1) sending an e-mail to the 0-level administrator
FPT_TST.1	TSF data integrity verification failure	The SSO server performs as follows: 1) sending an e-mail to the 0-level administrator The SSO agent performs as follows: 1) sending an e-mail to the 0-level administrator
FPT_TST.1	The validated cryptographic module's self-test failure	The SSO agent performs as follows: 1) at start-up: notifying the 0-level administrator by displaying error screen 2) during operation: notifying the 0-level administrator by sending an email The SSO agent performs as follows: 1) at start-up: notifying the 0-level administrator by displaying error screen 2) during operation: notifying the 0-level administrator by sending an email

5.1.1.2 FAU_GEN.1 Audit data generation

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

FAU_GEN.1.1

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

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

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

c) [Refer to the "auditable events" in <Table 5-3 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 PP/ST [Refer to the contents of "additional audit record" in <Table 5-3 Audit events> Audit events, [none]].

Table 5-3 Audit events

Security functional component	Auditable event	Additional audit record
FAU_ARP.1	Actions taken due to potential security violations	
FAU_SAA.1	Enabling and disabling of any of the analysis mechanisms, Automated responses performed by the tool	
FAU_STG.3	Actions taken due to exceeding of a threshold	
FAU_STG.4	Actions taken due to the audit storage failure	
FCS_CKM.1(1)(2)	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)(2)(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)	
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(1)(2)	All use of the authentication mechanism	
FIA_UAU.4	Attempts to reuse authentication data	
FIA_UID.2(1)(2)	All use of the administrator 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 All attempts at establishment of a user session	
FPT_TEE.1	Execution of the external entity tests and the results of the tests	

5.1.1.3 FAU_SAA.1 Potential violation analysis

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

FAU_SAA.1.1

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

FAU_SAA.1.2

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

a) Accumulation or combination of [refer to the “potential security violation” in <Table 5-2 Actions for potential security violation>] known to indicate a potential security violation;

b) [none].

5.1.1.4 FAU_SAR.1 Audit review

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

FAU_SAR.1.1

The TSF shall provide [**management console 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 **management console administrator** to interpret the information.

5.1.1.5 FAU_SAR.3(1) Selectable audit review (end-user log)

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

FAU_SAR.3.1

The TSF shall provide the ability to apply [the logical product(AND operation) among one of the items of the selection1, one of the items of the selection2, and one of the items of the ordering in <Table 5-4 Rules and methods for selecting and ordering audit data related to end-users>] of audit data **related to end-users** based on [the “method” in <Table 5-4 Rules and methods for selecting and ordering audit data related to end-users>].

Table 5-4 Rules and methods for selecting and ordering audit data related to end-users

category	items	method
Selection1	Audit log creation date	Choosing fixed period (today, yesterday, recent 1 week, recent 1 month) or user-defined period
Selection2	Audit log sequence number, user ID, user name, source IP, service name, log type, detailed information	Search for audit data where the substring matches the keyword If no keyword is specified, searching all logs.
Ordering	Audit log sequence number, log creation date and time, user ID, user name, source IP, service name, log type, detailed information	Sorting ascending or descending

5.1.1.6 FAU_SAR.3(2) Selectable audit review (administrator log)

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

FAU_SAR.3.1

The TSF shall provide the ability to apply [the logical product(AND operation) among one of the items of the selection1, one of the items of the selection2, and one of the items of the ordering in <Table 5-5 Rules and methods for selecting and ordering audit data related to management console administrator>] of audit data **related to management console administrators** based on [the "method" in <Table 5-5 Rules and methods for selecting and ordering audit data related to management console administrator>].

Table 5-5 Rules and methods for selecting and ordering audit data related to management console administrator

category	items	method
Selection1	Audit log creation date	Choosing between fixed (today, yesterday, recent 1 week, recent 1 month) or user-defined period
Selection2	Audit log sequence number, admin ID, admin name, source IP, log type, detailed information	Search for audit data where the substring matches the keyword If no keyword is specified, searching all logs.
Ordering	Audit log sequence number, log creation date and time, admin ID, admin name, source IP, log type, detailed information	Sorting ascending or descending

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

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

FAU_STG.3.1

The TSF shall [Notification to the **0-level administrator**, [none]] if the audit trail exceeds [the threshold set by the 0-level administrator (50% ~ 60%, default: 60%)].

5.1.1.8 FAU_STG.4 Prevention of audit data loss

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

FAU_STG.4.1

The TSF shall *"overwrite the oldest stored audit records"* and [Notification to the 0-level administrator] if the audit trail is full.

5.1.2 Cryptographic support (FCS)

5.1.2.1 FCS_CKM.1(1) Cryptographic key generation (KEK 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 [HMAC_SHA256] and specified cryptographic key sizes [128 bits] that meet the following: [PKCS #5(RFC 2898), TTAK.KO-12.0274].

5.1.2.2 FCS_CKM.1(2) Cryptographic key generation (secret keys other than KEK)

- 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 [HASH_DRBG SHA224] and specified cryptographic key sizes [128 bits, 256 bits] that meet the following: [TTAK.KO-12.0190].

5.1.2.3 FCS_CKM.2 Cryptographic key distribution

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

FCS_CKM.2.1

The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method [Offline distribution a file that stores the salt value for the same KEK derivation, DEK encrypted with KEK, and integrity verification key encrypted with KEK] that meets the following: [none].

5.1.2.4 FCS_CKM.4 Cryptographic key destruction

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

FCS_CKM.4.1

The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [zeroing by overwriting 5 times with 0x00] that meets the following: [none].

5.1.2.5 FCS_COP.1(1) Cryptographic operation (symmetric key)

- 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 [symmetric key cryptographic operations in <Table 5-6 Cryptographic operations (symmetric key)>] in accordance with a specified cryptographic algorithm [SEED (Mode=CBC)] and cryptographic key sizes [128 bits] that meet the following: [TTAS.KO-12.0004/R1, TTAS.KO-12.0025].

Table 5-6 Cryptographic operations (symmetric key)

Cryptographic key name	Cryptographic operation
Token En/Decryption Key	Encryption and decryption of authentication tokens
DEK	Encryption/decryption of TSF data and mutual authentication between TOE components except authentication token and encryption key
KEK	Encryption and decryption of the Token En/Decryption Key, DEK, Integrity Verification Key

5.1.2.6 FCS_COP.1(2) Cryptographic operation (HMAC)

- 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 [integrity verification of TSF and TSF data] in accordance with a specified cryptographic algorithm [HMAC-SHA256] and cryptographic key sizes [256 bits] that meet the following: [KS X ISO/IEC 9797-2].

5.1.2.7 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 [One-way encryption (HASH) of management console administrator's PW and end-user's PW] in accordance with a specified cryptographic algorithm [SHA256] and cryptographic key sizes [none] that meet the following: [ISO/IEC 10118-3].

5.1.2.8 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 [TTAK.KO-12.0190].

5.1.3 Identification and authentication (FIA)

5.1.3.1 FIA_AFL.1 Authentication failure handling

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

FIA_AFL.1.1

The TSF shall detect when "***0~1-level administrators*** *configurable positive numbers within [5 to 99]*" unsuccessful authentication attempts occur related to [0~3-level administrator's and end-user's authentication attempts].

FIA_AFL.1.2

When the defined number of unsuccessful authentication attempts has been *met* the TSF shall [disable authentication function for the management console administrator and the end-user for the amount of time 0~1-level administrator configurable positive integer within 5~90 minutes].

5.1.3.2 FIA_IMA.1 TOE Internal mutual authentication

- Hierarchical to: No other components.
- Dependencies: No dependencies.

FIA_IMA.1.1

The TSF shall perform mutual authentication between [the SSO server and the SSO agent] using the [two pass authentication mechanism] that meets the following [KS X ISO/IEC 9798 - 2].

5.1.3.3 FIA_SOS.1 Verification of secrets

- Hierarchical to: No other components.
- Dependencies: No dependencies.

FIA_SOS.1.1

The TSF shall provide a mechanism to verify that secrets meet [permission rules set by the administrator in FMT_PWD.1].

5.1.3.4 FIA_SOS.2 TSF Generation of secrets

- Hierarchical to: No other components.
- Dependencies: No dependencies.

FIA_SOS.2.1

TSF shall provide a mechanism to generate **an authentication token** that meet [a combination of business service ID, user ID, timestamp, user IP and token serial number].

FIA_SOS.2.2

TSF shall be able to enforce the use of TSF-generated **authentication token** for [authentication token-based end-user authentication].

5.1.3.5 FIA_SOS.3 Destruction of secrets (Extended)

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

FIA_SOS.3.1

The TSF shall destroy **authentication tokens** in accordance with a specified **authentication token** destruction method [zeroing by overwriting 5 times with 0x00] that meets the following: [none].

5.1.3.6 FIA_UAU.2(1) User authentication before any action (end-user)

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

FIA_UAU.2.1

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

5.1.3.7 FIA_UAU.2(2) User authentication before any action (management console administrator)

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

FIA_UAU.2.1

The TSF shall require each **management console administrator** to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that **management console administrator**.

5.1.3.8 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 [management console administrator's PW authentication mechanism, end-user's PW authentication mechanism and authentication token-based authentication mechanism].

5.1.3.9 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 [displaying the entered password masked (●), authentication success/failure indication] to the user while the authentication is in progress.

5.1.3.10 FIA_UID.2(1) User identification before any action (end-user)

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

FIA_UID.2.1

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

5.1.3.11 FIA_UID.2(2) User identification before any action (management console administrator)

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

FIA_UID.2.1

The TSF shall require each **management console administrator** to be successfully identified before allowing any other TSF-mediated actions on behalf of that **management console administrator**.

5.1.4 Security management (FMT)

5.1.4.1 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 management actions of*** the functions ["management functions" in <Table 5-7 Security management action for TSF>] to [**the 0~1-level administrator**].

Table 5-7 Security management action for TSF

Security functional component	Management function	Authorized role
FIA_SOS.1	Management of the metric used to verify the secrets	0~1-level administrator
FMT_PWD.1(Extended)	Management of ID and password configuration rules	0~1-level administrator

5.1.4.2 FMT_MTD.1 Management of TSF data

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

FMT_MTD.1.1

The TSF shall restrict the ability to ***manage*** the [TSF data in <Table 5-8 Security management action for TSF data>] to [**authorized role in <Table 5-8 Security management action for TSF data>**].

Table 5-8 Security management action for TSF data

Security functional component	TSF data	Authorized role
FAU_SAR.1	Maintenance (deletion, modification, addition) of the group of users with read access right to the audit records	0~1-level administrator
FAU_STG.3	Maintenance of the threshold	0-level administrator
FIA_AFL.1	Management of the threshold for unsuccessful authentication attempts	0~1-level administrator
FIA_UAU.2(1)	Management of the authentication data by the associated end-user	end-user
FIA_UAU.2(2)	Management of the authentication data by an administrator	0~2-level administrator
FIA_UID.2(2)	Management of the administrator identities	0~1-level administrator
FIA_UID.2(1)	Management of the end-user identities	0~2-level administrator
FMT_MOF.1	Management of the group of roles that can interact with the functions in the TSF	0~1-level administrator
FMT_MTD.1	Management of the group of roles that can interact with the TSF data	0~2-level administrator
FMT_SMR.1	Management of the group of users that are part of a role.	0~1-level administrator
FTA_SSL.5	Maximum allowed values for administrator and end-user inactivity periods	0~1-level administrator
FTA_TSE.1	Session establishment condition by authorized administrator	0~1-level administrator

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

- Hierarchical to: No other components.

- Dependencies:
  - FMT_SMF.1 Specification of Management Functions
  - FMT_SMR.1 Security roles

FMT_PWD.1.1

The TSF shall restrict the ability to manage the password of [functions of generating and changing end-user and 0~3-level administrator's PW] to [**the 0~1-level administrator**].

1. [Logical product(AND) of the acceptance criteria per each item in *<Table 5-9 Password combination rules and length>*]

2. [none]

FMT_PWD.1.2

The TSF shall restrict the ability to manage the ID of [none] to [the authorized administrator].

1. [none]

2. [none]

FMT_PWD.1.3

The TSF shall provide the capability for *changing the password when **the management console administrator and end-user** accesses for the first time*.

Table 5-9 Password combination rules and length

category	item	acceptance criteria
Combination rule	Special characters	Flag indicating inclusion which 0~1-level administrator can set - allowed characters: !\"#$%&'()*+,-./:;=?@[]^_`{|}~
Combination rule	Capital letters	Flag indicating inclusion which 0~1-level administrator can set
Combination rule	Small letters	Flag indicating inclusion which 0~1-level administrator can set
Combination rule	Numbers	Flag indicating inclusion which 0~1-level administrator can set
Combination rule	4 consecutive identical characters	Flag indicating inclusion which 0~1-level administrator can set
Combination rule	4 consecutive keyboard characters	Flag indicating inclusion which 0~1-level administrator can set
length	Minimum Password Length Limit	Flag indicating inclusion which 0~1-level administrator can set
Length	Minimum password length	- if the flag <Minimum Password Length Limit> is set, the minimum length is a positive number between 8 to 20 which 0~1-level administrator can define - if the flag <Minimum Password Length Limit> is not set, the minimum length is 8.
Length	Maximum password length	Under 20 characters
Change cycle	Password change cycle	change period that 0~1-level administrator can select - 1 month, 3 months, 6 months, 12 months, not used

5.1.4.4 FMT_SMF.1 Specification of Management Functions

- Hierarchical to: No other components.
- Dependencies: No dependencies.

FMT_SMF.1.1

The TSF shall be capable of performing the following management functions: [the list in <Table 5-7 Security management action for TSF>, the list in <Table 5-8 Security management action for TSF data>, and the list in <Table 5-9 Password combination rules and length>].

5.1.4.5 FMT_SMR.1 Security roles

- Hierarchical to: No other components.
- Dependencies:
  - FIA_UID.1 Timing of identification

FMT_SMR.1.1

The TSF shall maintain the roles [0~3-level administrators, end-user].

FMT_SMR.1.2

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

5.1.5 Protection of the TSF (FPT)

5.1.5.1 FPT_ITT.1 Basic Internal TSF data transfer protection

- Hierarchical to: No other components.
- Dependencies: No dependencies.

FPT_ITT.1.1

The TSF shall protect TSF data from *disclosure, modification* when it is transmitted between separate parts of the TOE.

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

- Hierarchical to: No other components.
- Dependencies: No dependencies.

FPT_PST.1.1

The TSF should protect the [password/email address/IP access control information of management console administrator and end-user, authentication token, token serial number, Integrity Verification Key, DEK, Token En/Decryption Key, TOE configuration values] stored in the repository, which is controlled by the TSF, from unauthorized *exposure and modification*.

5.1.5.3 FPT_TEE.1 Testing of external entities

- Hierarchical to: No other components.
- Dependencies: No dependencies.

FPT_TEE.1.1

The TSF shall run a suite of tests *at the request of* ***the 0~1-level administrators*** to check the fulfillment of [mail server, DBMS, web servers(WAS)].

FPT_TEE.1.2

If the test fails, the TSF shall [show alert window].

5.1.5.4 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 [*authentication token generation and verification function*].

FPT_TST.1.2

The TSF shall provide **during TOE initial start-up and the 0-level administrator** with the capability to verify the integrity of [*[<Table 5-10 TSF data subject to integrity* verification test*>*].

Table 5-10 TSF data subject to integrity verification test

Category	TSF data
SSO agent	The salt.dat file including follows: • 0-level administrator's ID/PW • Salt value for the KEK derivation • DEK, Integrity Verification Key • TOE configuration values such as business service ID, SSO server address
SSO server	Cryptographic keys and important TSF data • TOE configuration values • ID/password/email address/IP access control information of management console administrator and end-user, • cryptographic keys • authentication token, token serial number

FPT_TST.1.3

The TSF shall provide **during TOE initial start-up and the 0-level administrator** with the capability to verify the integrity of *TSF*.

5.1.6 TOE access (FTA)

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

- Hierarchical to:
  - FTA_MCS.1 Basic limitation on multiple concurrent sessions
- Dependencies:
  - FIA_UID.1 Timing of identification

FTA_MCS.2.1

The TSF shall restrict the maximum number of concurrent sessions that belong to the same user according to the rules [restriction to one for the maximum number of concurrent sessions for administrator management access session, prohibition of same administrator both concurrent connections of management access session and local access session, the Rules on the maximum number of concurrent sessions {for administrator HTTPS management access session, the maximum number of concurrent sessions for 0~2-level administrators is limited to 1, but the maximum number of concurrent sessions for the 3-level administrator is not limited according to <Table 5-11 Concurrent session limit rules for administrator's HTTPS connections>}]

FTA_MCS.2.2

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

Table 5-11 Concurrent session limit rules for administrator's HTTPS connections

Administrator level		Existing session		
		0~1-level	2-level	3-level
New session	0~1-level	Existing session terminated, New session allowed	Existing session terminated, New session allowed	concurrent session allowed
	2-level	Existing session maintained, New session prohibited	Existing session terminated, New session allowed	concurrent session allowed
	3-level	concurrent session allowed	concurrent session allowed	concurrent session allowed

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

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

FTA_SSL.5.1

The TSF shall *terminate* an interactive session after a [time interval of user inactivity which 0~1-level administrator can define].

5.1.6.3 FTA_TSE.1 TOE session establishment

- Hierarchical to: No other components.
- Dependencies: No dependencies.

FTA_TSE.1.1

The TSF shall be able to deny **management console administrator's management access** session establishment based on [connection IP, *None*].

5.2 Security assurance requirements

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

Table 5-12 Security assurance requirements

Security assurance class	Security assurance component
ASE: 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
ADV: Development	ADV_FSP.1 Basic functional specification
AGD: Guidance Documents	AGD_OPE.1 Operational user guidance
	AGD_PRE.1 Preparative procedures
ALC: Life-cycle support	ALC_CMC.1 Labelling of the TOE
	ALC_CMS.1 TOE CM coverage
ATE: Tests	ATE_FUN.1 Functional testing
	ATE_IND.1 Independent testing – conformance
AVA: Vulnerability assessment	AVA_VAN.1 Vulnerability survey

5.2.1 Security Target evaluation

5.2.1.1 ASE_INT.1 ST introduction

- Dependencies: No dependencies.
- Developer action elements
  - ASE_INT.1.1D The developer shall provide an ST introduction.
- Content and presentation elements
  - ASE_INT.1.1C The ST introduction shall contain an ST reference, a TOE reference, a TOE overview and a TOE description.
  - ASE_INT.1.2C The ST reference shall uniquely identify the ST.
  - ASE_INT.1.3C The TOE reference shall uniquely identify the TOE.
  - ASE_INT.1.4C The TOE overview shall 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.

5.2.1.2 ASE_CCL.1 Conformance claims

- Dependencies
  - ASE_INT.1 ST introduction
  - ASE_ECD.1 Extended components definition
  - ASE_REQ.1 Stated security requirements
- Developer action elements
  - ASE_CCL.1.1D The developer shall provide a conformance claim.
  - ASE_CCL.1.2D The developer shall provide a conformance claim rationale.
- Content and presentation elements
  - ASE_CCL.1.1C The conformance claim shall contain a CC conformance claim that identifies the version of the CC to which the ST and the TOE claim conformance.
  - ASE_CCL.1.2C The CC conformance claim shall describe the conformance of the ST to CC Part 2 as either CC Part 2 conformant or CC Part 2 extended.
  - ASE_CCL.1.3C The CC conformance claim shall describe the conformance of the ST to CC Part 3 as either CC Part 3 conformant or CC Part 3 extended.
  - ASE_CCL.1.4C The CC conformance claim shall be consistent with the extended components definition.
  - ASE_CCL.1.5C The conformance claim shall identify all PPs and security requirement packages to which the ST claims conformance.
  - ASE_CCL.1.6C The conformance claim shall describe any conformance of the ST to a package as either package-conformant or package-augmented.
  - ASE_CCL.1.7C The conformance claim rationale shall demonstrate that the TOE type is consistent with the TOE type in the PPs for which conformance is being claimed.
  - ASE_CCL.1.8C The conformance claim rationale shall demonstrate that the statement of the security problem definition is consistent with the statement of the security problem definition in the PPs for which conformance is being claimed.
  - ASE_CCL.1.9C The conformance claim rationale shall demonstrate that the statement of security objectives is consistent with the statement of security objectives in the PPs for which conformance is being claimed.
  - ASE_CCL.1.10C The conformance claim rationale shall demonstrate that the statement of security requirements is consistent with the statement of security requirements in the PPs for which conformance is being claimed.
- Evaluator action elements
  - ASE_CCL.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.1.3 ASE_OBJ.1 Security objectives for the operational environment

- Dependencies: No dependencies.
- Developer action elements
  - ASE_OBJ.1.1D The developer shall provide a statement of security objectives.
- Content and presentation elements
  - ASE_OBJ.1.1C The statement of security objectives shall describe the security objectives for the operational environment.
- Evaluator action elements
  - ASE_OBJ.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.1.4 ASE_ECD.1 Extended components definition

- Dependencies: No dependencies.
- Developer action elements
  - ASE_ECD.1.1D The developer shall provide a statement of security requirements.
  - ASE_ECD.1.2D The developer shall provide an extended components definition.
- Content and presentation elements
  - ASE_ECD.1.1C The statement of security requirements shall identify all extended security requirements.
  - ASE_ECD.1.2C The extended components definition shall define an extended component for each extended security requirement.
  - ASE_ECD.1.3C The extended components definition shall describe how each extended component is related to the existing CC components, families, and classes.
  - ASE_ECD.1.4C The extended components definition shall use the existing CC components, families, classes, and methodology as a model for presentation.
  - ASE_ECD.1.5C The extended components shall consist of measurable and objective elements such that conformance or nonconformance to these elements can be demonstrated.
- Evaluator action elements
  - ASE_ECD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
  - ASE_ECD.1.2E The evaluator shall confirm that no extended component can be clearly expressed using existing components.

5.2.1.5 ASE_REQ.1 Stated security requirements

- Dependencies
  - ASE_ECD.1 Extended components definition
- Developer action elements
  - ASE_REQ.1.1D The developer shall provide a statement of security requirements.
  - ASE_REQ.1.2D The developer shall provide a security requirements rationale.
- Content and presentation elements

- ASE_REQ.1.1C The statement of security requirements shall describe the SFRs and the SARs.
- ASE_REQ.1.2C All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.
- ASE_REQ.1.3C The statement of security requirements shall identify all operations on the security requirements.
- ASE_REQ.1.4C All operations shall be performed correctly.
- ASE_REQ.1.5C Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.
- ASE_REQ.1.6C The statement of security requirements shall be internally consistent.

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

5.2.1.6 ASE_TSS.1 TOE summary specification

- Dependencies
  - ASE_INT.1 ST introduction
  - ASE_REQ.1 Stated security requirements
  - ADV_FSP.1 Basic functional specification
- Developer action elements
  - ASE_TSS.1.1D The developer shall provide a TOE summary specification.
- Content and presentation elements
  - ASE_TSS.1.1C The TOE summary specification shall describe how the TOE meets each SFR.
- Evaluator action elements
  - ASE_TSS.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
  - ASE_TSS.1.2E The evaluator shall confirm that the TOE summary specification is consistent with the TOE overview and the TOE description.

5.2.2 Development

5.2.2.1 ADV_FSP.1 Basic functional specification

- Dependencies: No dependencies.
- Developer action elements
  - ADV_FSP.1.1D The developer shall provide a functional specification.
  - ADV_FSP.1.2D The developer shall provide a tracing from the functional specification to the SFRs.
- Content and presentation elements
  - ADV_FSP.1.1C The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.

- ADV_FSP.1.2C The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.
- ADV_FSP.1.3C The functional specification shall provide rationale for the implicit 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

5.2.3.1 AGD_OPE.1 Operational user guidance

- Dependencies
  - ADV_FSP.1 Basic functional specification
- Developer action elements
  - AGD_OPE.1.1D The developer shall provide operational user guidance.
- Content and presentation elements
  - AGD_OPE.1.1C The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.
  - AGD_OPE.1.2C The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.
  - AGD_OPE.1.3C The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.
  - AGD_OPE.1.4C The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.
  - AGD_OPE.1.5C The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.
  - AGD_OPE.1.6C The operational user guidance shall, for each user role, describe the security measures to be followed in order to 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.

5.2.3.2 AGD_PRE.1 Preparative procedures

- Dependencies: No dependencies.
- Developer action elements
  - AGD_PRE.1.1D The developer shall provide the TOE including its preparative procedures.
- Content and presentation elements
  - AGD_PRE1.1C The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.
  - AGD_PRE1.2C The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.
- Evaluator action elements
  - AGD_PRE.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
  - AGD_PRE.1.2E The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

5.2.4 Life-cycle support

5.2.4.1 ALC_CMC.1 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.

5.2.4.2 ALC_CMS.1 TOE CM coverage

- Dependencies: No dependencies.
- Developer action elements
  - ALC_CMS.1.1D The developer shall provide a configuration list for the TOE.
- Content and presentation elements
  - ALC_CMS.1.1C The configuration list shall include the following: the TOE itself; and the evaluation evidence required by the SARs.
  - ALC_CMS.1.2C The configuration list shall uniquely identify the configuration items.
- Evaluator action elements
  - ALC_CMS.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.5 Tests

5.2.5.1 ATE_FUN.1 Functional testing

- Dependencies
  - ATE_COV.1 Evidence of coverage
- Developer action elements
  - ATE_FUN.1.1D The developer shall test the TSF and document the results.
  - ATE_FUN.1.2D The developer shall provide test documentation.
- Content and presentation elements
  - ATE_FUN.1.1C The test documentation shall consist of test plans, expected test results and actual test results.
  - ATE_FUN.1.2C The test plans shall identify the tests to be performed and describe the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests.
  - ATE_FUN.1.3C The expected test results shall show the anticipated outputs from a successful execution of the tests.
  - ATE_FUN.1.4C The actual test results shall be consistent with the expected test results.
- Evaluator action elements
  - ATE_FUN.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.5.2 ATE_IND.1 Independent testing - conformance

- Dependencies
  - ADV_FSP.1 Basic functional specification
  - AGD_OPE.1 Operational user guidance
  - AGD_PRE.1 Preparative procedures
- Developer action elements
  - ATE_IND.1.1D The developer shall provide the TOE for testing.
- Content and presentation elements
  - ATE_IND.1.1C The TOE shall be suitable for testing.
- Evaluator action elements
  - ATE_IND.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
  - ATE_IND.1.2E The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

5.2.6 Vulnerability assessment

5.2.6.1 AVA_VAN.1 Vulnerability survey

- Dependencies
  - ADV_FSP.1 Basic functional specification

- AGD_OPE.1 Operational user guidance
- AGD_PRE.1 Preparative procedures

- Developer action elements
  - AVA_VAN.1.1D The developer shall provide the TOE for testing
- Content and presentation elements
  - AVA_VAN.1.1C The TOE shall be suitable for testing.
- Evaluator action elements
  - AVA_VAN.1.1E The evaluator shall confirm that the information provided meets all requirements for content and 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 rationale of security functional requirements

The following table shows dependency of security functional requirement.

Table 5-13 Rationale for the dependency of the security functional requirements

No.	Security functional requirements	Dependency
1	FAU_ARP.1	FAU_SAA.1
2	FAU_GEN.1	FPT_STM.1
3	FAU_SAA.1	FAU_GEN.1
4	FAU_SAR.1	FAU_GEN.1
5	FAU_SAR.3(1)(2)	FAU_SAR.1
6	FAU_STG.3	FAU_STG.1
7	FAU_STG.4	FAU_STG.1
8	FCS_CKM.1(1)(2)	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4
9	FCS_CKM.2	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4
10	FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]
11	FCS_COP.1(1)(2)(3)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4
12	FCS_RBG.1	-
13	FIA_IMA.1	-
14	FIA_AFL.1	FIA_UAU.1
15	FIA_SOS.1	-
16	FIA_SOS.2	-
17	FIA_SOS.3	FIA_SOS.2

18	FIA_UAU.2(1)(2)	FIA_UID.1
19	FIA_UAU.4	-
20	FIA_UAU.7	FIA_UAU.1
21	FIA_UID.2(1)(2)	-
22	FMT_MOF.1	FMT_SMF.1 FMT_SMR.1
23	FMT_MTD.1	FMT_SMF.1 FMT_SMR.1
24	FMT_PWD.1	FMT_SMF.1 FMT_SMR.1
25	FMT_SMF.1	-
26	FMT_SMR.1	FIA_UID.1
27	FPT_ITT.1	-
28	FPT_PST.1	-
29	FPT_TEE.1	-
30	FPT_TST.1	-
31	FTA_MCS.2	FIA_UID.1
32	FTA_SSL.5	FIA_UAU.1 or No dependencies
33	FTA_TSE.1	-

FAU_GEN.1 has the dependency on FPT_STM.1, but in the case of TOE of this Security Target, since the function is supported by the operating environment, the security objective (OE.TIMESTAMP) for the operating environment is added, therefore the dependency is satisfied.

FAU_STG.3 and FAU_STG.4 have the dependency on FAU_STG.1, but in the case of TOE of this Security Target, since the function is supported by the operating environment (DBMS), the security objective (OE. DBMS) for the operating environment is added, therefore the dependency is satisfied.

FIA_UAU.2, FMT_SMR.1, and FTA_MCS.2 have dependency on FIA_UID.1, but in this ST, FIA_UID.2, which has hierarchical to this SFR, is used, therefore the dependency is satisfied.

FIA_AFL.1, FIA_UAU.7 and FTA_SSL.5 have dependency on FIA_UAU.1, but in this ST, FIA_UAU.2, which has hierarchical to this SFR, is used, therefore the dependency is satisfied.

5.3.2 Dependency rationale of security assurance requirements

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 ST since it is not necessarily required to show the correspondence between the tests and the TSFIs.

6 TOE Specification summary

6.1 Security Audit (FAU)

6.1.1 Audit record generation

- *Related SFR: FAU_GEN.1*

  Since the SSO server handles the audit-related function of the TSF, the SSO server creates audit records about 'start-up' event, 'shut-down' event and other auditable events described in <Table 5-3 Audit events>. And the SSO server stores audit records in DBMS(MariaDB) which is the audit data storage.

  When creating audit records, the information to be recorded is date and time of the event, type of event, subject identity(if applicable) and outcome (success or failure) of the event, and for some audit events described in <Table 5-3 Audit events>, additional audit record will be included.

6.1.2 Audit Review

- *Related SFR: FAU_SAR.1, FAU_SAR.3(1)(2)*

  All administrators can review audit logs created by TSF through the management console.

  Audit log to review can be chosen based on the period of the audit log creation time. Or with the AND combination of (1) one keyword from creation sequence number, user id, user name, source IP, service name, log type, or specific information / (2) period search condition based on creation time, audit log can be chosen.

  Descending order sorting or ascending order sorting can also be done based on the sequence number of audit log, log creation time, user id, user name, source IP, service name, log type, or specific information.

  There are 2 types of audit logs: administrator log and user log.

6.1.3 Security alarms

- *Related SFR: FAU_SAA.1, FAU_ARP.1*

  The TSF detects potential violation by analyzing audit data of potential security violation events described in <Table 5-2 Actions for potential security violation>. For FIA_UAU.2 authentication mechanism failure events, when events are cumulated as many as the value of TSF data that can be composed by 0~1-level administrator, the TOE detects these as a potential security violation.

  Other events are detected as a potential violation when it happens at once. According to the <Table 5-2 Actions for potential security violation>, the actions against each potential security violation will be conducted.

6.1.4 Protection of audit data

- *Related SFR: FAU_STG.3, FAU_STG.4*

  The thread that runs every one minute after the SSO server starts, checks whether the size of audit trail exceeds the threshold of storage capacity (disk partition to store DB). And if the size exceeds the

threshold, the TOE notices 0-level administrator via e-mail. The value range to be set for threshold is 50~60(%), and can be configured by 0-level administrator.

And, if the limit of the storage capacity configurable by 0-level administrator is exceeded, the audit trail is considered to be saturated. The thread that runs every minute after the SSO server starts, checks whether the audit trail is saturated. If it does, the thread sends an e-mail to the 0-level administrator and deletes the DB partition with the oldest audit records until the size of the audit trail is under the threshold. The value range of the storage capacity limit is (the value of set threshold + 1) ~ 70(%).

6.2 Cryptographic support (FCS)

6.2.1 Cryptographic key generation and random bit generation

■ *Related SFR: FCS_CKM.1(1)(2), FCS_RBG.1*

<Table 6-1 Cryptographic key generation> describes the type, purposes, key length, key generation algorithm and reference standards, and the way of generation of cryptographic key required for TOE.

Table 6-1 Cryptographic key generation

Cryptographic key	Key length	Purposes	Cryptographic key generation algorithm and reference standards
KEK	128 bits	En/decryption of DEK, Token En/Decryption Key, Integrity Verification Key	• The KEK generation method by deriving it from the password follows PKCS #5(RFC 2898), TTAK.KO-12.0274. • Pseudo random function: HMAC-SHA256 • Salt value: a random number of 128 bits generated by using the approved random bit generator (HASH_DRBG SHA224) of the validated cryptographic module (CIS-CC v3.3) • Iteration count: 1000 times
DEK	128 bits	En/decryption of TSF data and mutual authentication except authentication token	• A secret key for symmetric block encryption algorithm (SEED), which is a random number of 128 bits generated by using the approved random bit generator (HASH_DRBG SHA224 which meets TTAK.KO-12.0190) of the validated cryptographic module (CIS-CC v3.3)
Token En/Decryption Key	128 bits	En/decryption of authentication token	• A secret key for symmetric block encryption algorithm (SEED), which is a random number of 128 bits generated by using the approved random bit generator (HASH_DRBG SHA224 which meets TTAK.KO-12.0190) of the validated cryptographic module (CIS-CC v3.3)
Integrity Verification Key	256 bits	hmac secret key for TSF, TSF data integrity verification	• A secret key for HMAC, which is a random number of 256 bits generated by using the approved random bit generator (HASH_DRBG SHA224 which meets TTAK.KO-12.0190) of the validated cryptographic module (CIS-CC v3.3)

6.2.2 Cryptographic key distribution

- *Related SFR: FCS_CKM.2*

When the administrator adds a business service with the management console, the SSO server stores the salt value for the KEK generation, the DEK and the integrity verification key encrypted with the KEK in salt.dat file and automatically exports it when the download button is clicked. The salt.dat file is distributed offline to the SSO agent with a USB memory, etc. The SSO agent then automatically imports the salt.dat file at the start-up phase.

To guarantee the integrity of TSF data stored in salt.dat file, an hmac value generated by hmac-sha256 operation of validated cryptographic module “CIS-CC v3.3” using the integrity verification key is stored together in salt.dat file.

6.2.3 Cryptographic key destruction

- *Related SFR: FCS_CKM.4*

SSO server and SSO agent generate KEK in PKCS # 5 method at every startup, and then they load and decrypt the DEK and the integration verification key which are encrypted with the KEK and stored in DBMS(for the SSO server) or the salt.dat file (for the SSO agent).

The plain type KEK, DEK and integrity verification key are saved in the memory with the internally implemented encoding technique. When they are used in the cryptographic key distribution and cryptographic operation, they are copied to the temporary variable and decoded. The used plaintext cryptographic key in the temporary variable is zeroized by overwriting 5 times with 0x00 and deleted from memory. When the TOE shuts down, the KEK, DEK, and integrity verification key in the memory with the internally implemented encoding technique are zeroized by overwriting 5 times with 0x00 and deleted from memory.

The password used to derive the KEK is zeroized by overwriting 5 times with 0x00 and deleted from the memory.

When an authentication token is created, the token en/decryption key is also generated and used for encrypting the token. The plain token en/decryption key in the memory is zeroized and deleted, and the key encrypted with the KEK is stored in the DB. When the authentication token is verified, the encrypted token en/decryption key stored in the DB is loaded into memory and decrypted with the KEK, and after decrypting the encrypted authentication token, the plain token en/decryption key and the KEK are zeroized and deleted from memory.

6.2.4 Cryptographic operations

- *Related SFR: FCS_COP.1(1)(2)(3)*
  - Symmetric key (KEK, DEK, Token En/Decryption Key)

The TSF uses cryptographic algorithms SEED (Mode = CBC) conforming to TTAS.KO-12.0004 / R1 and TTAS.KO-12.0025 to perform the cryptographic operation of <Table 5-6 Cryptographic operations (symmetric key)>. The TSF generates a 128-bit random number using HASH-DRBG (SHA224) of the validated cryptographic module (CIS-CC v3.3), which is a validated random number generator conforming to TTAK.KO-12.0190, and uses it as IV required for CBC mode of SEED algorithm.

  - HMAC (Integrity Verification Key)

The TSF performs the integrity verification of the TSF and TSF data by using the encryption algorithm HMAC-SHA256 conforming to KS X ISO / IEC 9797-2 with a cryptographic key of 256 bits length. (The validated cryptographic module, CIS-CC v3.3 is used.)

- HASH

Using one-way cryptographic algorithm SHA256 corresponding to ISO/IEC 10118-3, login PW of management console administrator and end-user is encrypted. In order to generate different ciphertexts each time, even if the same PW is used, the combination of the time information (timestamp) at which the PW is changed and the ID of the corresponding user is used as the salt value.

- Validated cryptographic module

Cryptographic module name & version	Validation number	Validation date	Developer
CIS-CC v3.3	CM-145-2023.11	2018-11-07	Penta Security System Inc.

6.3 Identification and authentication (FIA)

6.3.1 TOE Internal mutual authentication (between SSO server and SSO agent)

- *Related SFR: FIA_IMA.1*

For the mutual authentication between the SSO server and the SSO agent, which are physically separated TOE components, the two pass authentication mechanism conforming to the KS X ISO/IEC 9798-2 standard is applied.

6.3.2 Identification and authentication of the administrators

- *Related SFR: FIA_AFL.1, FIA_UAU.2(2), FIA_UAU.4, FIA_UAU.7, FIA_UID.2(2)*

All management console administrators can log in to the management console of SSO server with their ID and password. When an administrator initially logs in the TOE, he/she can use the default password (for the 0-level administrator) or the temporary password (for 1~3-level administrators) and if he/she succeeds to log in, a screen to mandatorily change the password will pop up. Without changing the password, the administrator will have no access to other management console functions.

If the management console administrator incorrectly enters the password continuously, it will not be able to log in with that ID for a certain period of time. Failure allowance count and account lockout time can be set by 0~1-level administrators in the management console. The default setting for the failure allowance count is five, and can be changed to a positive number between 5 and 99. The default setting for account lockout time is 5 minutes, and can be changed to a positive number between 5 and 90.

The passwords that administrators enter when logging in or changing passwords are masked (●) and displayed on the screen. When login fails, specific reasons for failure are not displayed.

The TSF issues a session ID when the administrator's ID/PW-based authentication is performed. The session ID is a UUID (Universal Unique ID), which is randomly generated every time a request is made. If the session ID submitted by the client at each request does not match the session ID stored in the session, the session ID is considered to have been reused and the session is blocked.

6.3.3 Identification and authentication of the end-users

- *Related SFR: FIA_AFL.1, FIA_SOS.2, FIA_SOS.3, FIA_UAU.2(1), FIA_UAU.4, FIA_UAU.7, FIA_UID.2(1)*

End-users use TOE Single Sign On service to log into business server. The end-user identification and authentication procedure can be grouped into the initial authentication phase using the ID/PW, and the

token-based authentication phase that accesses the business system using the token issued during the initial authentication procedure.

The execution procedure of the initial authentication phase is as follows. (1) When the user accesses the business server, the SSO agent redirects to the SSO server. Then, the user enters ID / PW in the login page of the SSO server to request a user login. (2) SSO server performs login verification using user information stored in DBMS. The SSO server issues an authentication token and passes it to the end-user's browser if the login validation result is valid, (3) then redirects to the SSO agent. (4) The SSO agent requests authentication token verification from the SSO server. (5) The SSO server validates the authentication token and returns the result. At this time, the authentication token is updated. The SSO agent allows the user to use the business service if the authentication token verification succeeds (Business service login success).

The token-based authentication step is performed only when the authentication token is successfully issued through the initial authentication step. (6) When the user accesses the business server, the SSO agent redirects to the SSO server. Since the SSO server already has an SSO session for the user, it extracts the authentication token from the SSO session and sends it to the user. The user sends an authentication token verification request to the SSO agent. (7) The SSO agent requests authentication token verification from the SSO server. (8) The SSO server validates the authentication token and returns the result. At this time, the authentication token is updated. The SSO agent allows the user to use the business service if the authentication token verification succeeds (Business service login success).

The authentication token is a combined string of business service ID, user ID, token creation time, user IP, and token serial number. A token en/decryption key is generated each time an authentication token is generated and updated. The combined string of the plain authentication token and the HMAC (HMAC-SHA256, 256bits) value for the authentication token, is encrypted with the token en/decryption key by the SEED (CBC mode, 128 bits) algorithm, and the authentication token ciphertext is used for communication and storage.

The authentication token ciphertext is zeroized by overwriting 5 times with 0x00 and deleted from the memory after transmission and / or DB storage in the authentication token issuance processing and the token verification request processing. Therefore, there is no authentication token ciphertext in the memory when the user session is terminated and the SSO server is shut down.

The authentication token plaintext is temporarily in memory only upon generation and verification of the authentication token. After the generation and verification of the authentication token is completed, the authentication token plaintext is zeroized by overwriting 5 times with 0x00 and deleted from the memory.

The HMAC value used in generating the authentication token ensures the integrity of the authentication token, and the SEED algorithm ensures the confidentiality of the authentication token. These cryptographic operations and the generation of the token en/decryption key are performed using the validated cryptographic module CIS-CC v3.3.

The token serial number is incremented by 1 for every verification since the authentication token is generated. When the SSO server receives the authentication token verification request, it verifies the timeliness and uniqueness of the authentication token using the authentication token creation time information and the token serial number included in the authentication token. If the difference between the authentication token creation time and the verification request time is longer than a set time or if the token serial number included in the authentication token does not match the token serial number of the corresponding SSO session stored in the SSO server, the SSO server prevents an attempt to reuse it by denying the authentication token verification.

The TSF issues a session ID when the end-user's ID/PW-based authentication is performed. The session ID is a UUID (Universal Unique ID), which is randomly generated every time a request is made. If the

session ID submitted by the client at each request does not match the session ID stored in the session, the session ID is considered to have been reused and the session is blocked.

If an end-user incorrectly enters the password continuously, it will not be able to log in with that ID for a certain period of time. Failure allowance count and account lockout time can be set by 0~1-level administrators in the management console. The default setting for the failure allowance count is five, and can be changed to a positive number between 5 and 99. The default setting for account lockout time is 5 minutes, and can be changed to a positive number between 5 and 90.

The passwords that end-users enter when logging in or changing passwords are masked (●) and displayed on the screen. When login fails, specific reasons for failure are not displayed.

6.4 Security management (FMT)

6.4.1 Management of security functions behaviour

- *Related SFR: FMT_MOF.1, FMT_SMF.1, FMT_SMR.1*

  0~1-level administrators can perform the security management actions of <Table 5-7 Security management action for TSF> through the management console.

6.4.2 Management of TSF data

- *Related SFR: FMT_MTD.1, FMT_SMF.1, FMT_SMR.1*

  The 0~2-level administrators can conduct management actions in <Table 5-8 Security management action for TSF data> through management console. The scope of the administrator with the management function is different for each TSF data. For more information, see <Table 5-8 Security management action for TSF data>.

  3-level administrators have only the right to view audit logs among the TSF data.

6.4.3 Management of ID and password

- *Related SFR: FMT_PWD.1, FMT_SMF.1, FMT_SMR.1, FIA_SOS.1*

  0~1-level administrators can add / delete / modify 1~3-level administrators' account in the management console.

  For the 0-level administrator, there is only one administrator account with a fixed ID ('adm'). When an administrator first accesses the management console, the administrator is logged in with the default password (for the 0-level administrator) or temporary password (for 1~3-level administrators). A screen for prompting the administrator to change the password is displayed immediately after login is successful. Without changing the password, the administrator will not be able to use other features of the management console.

  The administrator's password must conform to the password combination rules and length (see <Table 5-9 Password combination rules and length>.) set by 0~1-level administrators. Also, these rules must be conformed when the TSF automatically generating a temporary password to create a new administrator account, or when the administrator changing the password directly.

  0~2-level administrators can add / delete / modify end-user groups and accounts in the management console.

The end-user group is only used to display a list of users belonging to a specific group in the management console, and there's no difference in rights among the users belonging to different groups.

The password of end-users shall conform to the length and combination rules of the user password set by the 0~1-level administrators in management console (refer to <Table 5-9 Password combination rules and length>). Also, these rules must be conformed when the TSF automatically generating a temporary password to create a new end-user account, or when the user changing the password directly.

When an end-user logs in with a temporary password for the first time to access to a business service, a screen prompting the user to change the password is displayed immediately after the successful login. If the password is not changed, the end-user can not log in to the business service.

6.5 TSF protection (FPT)

6.5.1 Internal TSF data transfer protection

- *Related SFR: FPT_ITT.1*

All TSF data transmitted between SSO agent and SSO server is given an hmac value generated by the integrity verification key and encrypted by DEK. The internally transmitted TSF data includes authentication token verification request/result and SSO agent's request to store audit data. The hmac-sha256 operation with 256-bit length the integrity verification key is used to ensure the integrity of internally transmitted TSF data, and SEED operation (CBC mode) with 128-bit length DEK is used to ensure the confidentiality of the data. In addition to all cryptographic operations, random number generation for generating IV values needed for CBC mode uses HASH-DRBG (sha224) of CIS-CC v3.3, the validated cryptographic module.

6.5.2 Protection of stored TSF data

- *Related SFR: FPT_PST.1*

The TSF data required for the operation of the SSO server is stored in the external entity DBMS (MariaDB). Important TSF data stored in DBMS is encrypted by the validated cryptographic module (CIS-CC v3.3) to ensure confidentiality. And hmac values (HMAC-SHA256 with 256-bit length key) for each TSF data are stored together to ensure integrity of TSF data.

User data are also stored in DBMS, but they are not encrypted. The user data are not related to the TSF such as the name, employee number, and telephone number of the administrators and the end-users.

For the SSO agent, the TSF data is stored in the salt.dat file.

TSF data protection mechanisms are summarized in <Table 6-2 Protection mechanism of the stored TSF data>.

Table 6-2 Protection mechanism of the stored TSF data

TSF data	Protection mechanism
0~3-level administrator and end-user PW	• Applies salt at SHA256
KEK	• Generated through PKCS#5 at every start-up and is not stored in DB • Plain KEK issued is converted according to the internally implemented encoding technique and stored in the memory.
Integrity Verification Key	• Encrypted with KEK (SEED, CBC mode, 128bits) containing hmac value and stored in DB

	• Plain Integrity Verification Key (256bits) issued is converted to 512 bits data according to the internally implemented encoding technique and stored in the memory.
DEK	• Encrypted with KEK (SEED, CBC mode, 128bits) containing hmac value and stored in DB • Plain DEK (128 bits) issued is converted to 512 bits data according to the internally implemented encoding technique and stored in the memory.
Token En/Decryption Key	• Encrypted with KEK (SEED, CBC mode, 128bits) and stored in DB
Authentication token	• Hmac value is appended to the plain authentication token and they are encrypted with the token en/decryption key (SEED, CBC mode, 128bits).
Important TSF data(TOE setting, administrator/user e-mail, IP access control)	• Encrypted with DEK (SEED, CBC mode, 128bits) containing hmac value and stored in DB.

6.5.3 Testing of external entities

■ *Related SFR: FPT_TEE.1*

When the 0~1-level administrator requests, DBMS, web server (WAS) and mail server which are the external entities, are tested for availability. If the test fails, a notification message is displayed on the management console screen if possible.

The TSF sends a sample e-mail to the mail server at the request of the 0~1-level administrator to determine whether the mail server is available and displays the result in a notification message on the management console screen.

The TSF determines whether the DBMS is available by attempting to query the DBMS at the request of the 0~1-level administrator, and displays the result in a notification message on the management console screen.

The TSF determines whether the WAS is available by checking the running WAS process with the specific execution option of the SSO server at the request of the 0~1-level administrator, and displays the result in a notification message on the management console screen. If the WAS, the operating environment of the management console, is not available, the WAS can not process the availability test request from the browser of the administrator PC. Therefore, the availability test can not be performed by the TSF and the result also can not be displayed on the management console screen.

When the administrator makes an availability test request, the availability test for DBMS and WAS is performed sequentially. If the DBMS availability test fails, normal operation of the TOE is not possible. Then, the availability test for WAS is not meaningful, so it is not performed.

6.5.4 TSF testing and integrity verification

■ *Related SFR: FPT_TST.1*

When the SSO server and SSO agent are started, the pre-operational self-tests of the validated cryptographic module are performed. And the conditional self-test of the validated cryptographic module is performed during the operation of the TSF. The TSF shall notify the 0-level administrator of the failure of the self-test of the validated cryptographic module by displaying an error screen or sending an e-mail.

The SSO server performs self-tests of the authentication token generation and verification process at startup and periodically (every 6 hours).

The self-test is performed using the Known Answer Test. An authentication token is generated with fixed input values, and the generated authentication token is verified. If the output values of the token verification match the input values, the self-test is successful. In other cases, that is, if the input / output values do not match, or if the cryptographic operations performed in the self-test process fail, the self-test fails.

The SSO server verifies the integrity of all TSF and important TSF data of the SSO server at start-up and at the request of the 0-level administrator.

When the SSO server performs the self-test and integrity verification tests, it records the results in the administrator log and notifies the 0-level administrator by email if the tests fail.

The SSO agent performs integrity verification tests on all TSF and TSF data (salt.dat file) at startup and periodically (every 6 hours). And it records the results in the administrator log and notifies the 0-level administrator by email if the tests fail.

<Table 5-10 TSF data subject to integrity verification test> shows the TSF data to be subject to integrity verification. In the case of the TSF, integrity verification tests are performed on the entire TSF.

6.6 TOE access (FTA)

6.6.1 Administrator TOE access restrictions

- *Related SFR: FTA_MCS.2, FTA_SSL.5, FTA_TSE.1*

0~3-level administrators can access to the SSO server through HTTPS communication. By default, the number of concurrent sessions between 0~2-level administrators and between same account sessions is max. 1 except for 3-level administrators. Concurrent sessions are restricted according to <Table 5-11 Concurrent session limit rules for administrator's HTTPS connections>.

0~3-level administrator's session to the SSO server is automatically logged out and the session is terminated when the admin session timeout period (default 10 minutes, 5 ~ 720 minutes can be set by the 0~1-level administrator) has elapsed. When the admin session ends automatically, it automatically switches to the login screen.

The administrator's management access session is only allowed on the access permission IP set in the management console. When an administrator attempts to log in, the TSF performs administrator identification using the administrator information of the DB, inquires the IP of the client (administrator PC) from the administrator access session, compares it with the access permission IP list of the administrator information, if not, disconnect the connection.

6.6.2 End-user TOE access restrictions

- *Related SFR: FTA_SSL.5*

A session in which an end-user logs in to the SSO server with initial authentication is terminated after the session timeout period elapses since the last authentication (initial authentication or token-based authentication). The user session timeout period can be set by the 0~1-level administrator in the management console. The range of the timeout period is 5 ~ 10,080 minutes and the default value is 10 minutes.