Ubintis Lab

Pass-Ni SSO v5.0

Security Target

V1.0 R3

UbiNtisLab Co., Ltd

Revision History

Ver	Data	Detail	Author
R1	2022-11-22	Initial release	Research Institute
R2	2023-11-22	ST Updated	Research Institute
R3	2023-12-28	TOE Updated (v5.0.002)	Research Institute

Table of Contents

1. ST Introduction

This chapter introduces the Security Target (ST) of Pass-Ni SSO v5.0 of UbiNtisLab Co., Ltd.

1.1 ST Reference

Item	Specification
Title	Pass-Ni SSO v5.0 Security Target
Version	V1.0 R3
Author	UbiNtisLab Co., Ltd
Publication Date	2023-12-28
Configuration Management No.	PassNi-SSO-v5.0-ST-V1.0.R3
Common Criteria	PassNi-SSO-v5.0-ST-V1.0.R3.docx
Protection Profile	Common Criteria for Information Technology Security Evaluation V3.1R5 (CC V3.1 r5)
Evaluation Assurance Level	National Protection Profile for Single Sign-On V1.1 (KECS-PP-0822a-2017)
Title	EAL1+(ATE_FUN.1)

[Table 1-1] ST Reference

1.2 TOE Reference

Item		Specification
TOE		Pass-Ni SSO v5.0
TOE Version		v5.0.002
TOE Components	SSO Server	Pass-Ni SSO Server v5.0.002
	SSO Agent	Pass-Ni SSO Agent v5.0.002
Guidance Document	Preparative Procedures	Pass-Ni SSO v5.0 Preparative Procedures V1.0 R3
	Operational Guidance	Pass-Ni SSO v5.0 Operational Guidance V1.0 R3

Developer	UbiNtisLab Co., Ltd

[Table 1-2] TOE Reference

1.3 TOE Overview

This section prescribes the usage of the TOE and major security features. It also identifies types of the TOE and identifies software, hardware and firmware required by the TOE but not the TOE.

1.3.1 TOE Types

The TOE is offored in the form of software which is 'Single Sign On (SSO)' that allows access to various application servers (business systems) through a single user login.

The TOE component comprised of the SSO server that performs functions such as processing user login, issuing authentication token and managing policy, and the SSO Agent that is installed in each business system and verifies the validity of the authentication token through interworking with the SSO Server. The SSO agent is provided as 'API' consisting of library files.

The SSO server and SSO agent, which are TOE components, use verified cryptographic modules whose safety and implementation suitability have been verified for cryptographic operations such as encryption key generation, encryption and decryption, integrity verification, and encrypted communication between components.

The TOE uses the following validated cryptographic module.

Item	Specification
Cryptographic module name	Pass-Ni Crypto V2.0
Validation number	CM-215-2027.10
Validation date	2022-10-04
Expiration date	2027-10-04
Developer	UbiNtisLab Co., Ltd

[Table 1-3] Cryptographic Module Reference

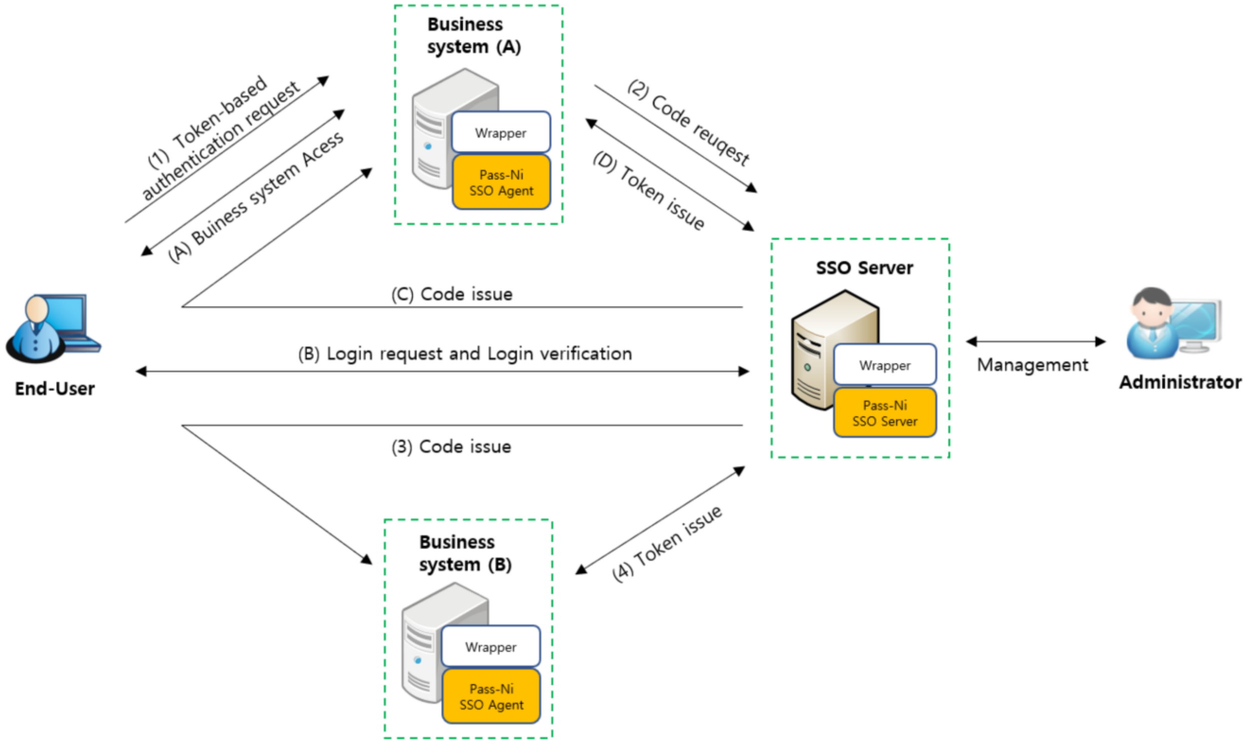
1.3.2 TOE Usages and Major Security Features

The TOE is an 'Single Sign On (SSO)' that is used for the purpose of providing a user with services from various application servers (business systems) without additional login by single login. The TOE provides users with access to information of various business systems through single authentication.

The major features of the TOE are to issue, store, verify, and revoke the authentication token. It issues an authentication token when the user, who requests the login, is regarded as valid by identifying and authenticating. Then, when the user accesses the other business system, the access of the user is controlled through validation of the authentication token. In the initial authentication phase, the TOE performs ID / PW authentication for doing identification and authentication functionality.

TOE security functions include **the security audit function** that manages and records major cases as audit data, **the identification and authentication function** for users, **TSF protection functions** such as TSF data protection function and TSF self test. It also includes **the cryptographic support function** for performing cryptographic key management and cryptographic operations, **the security management function** for security policy and environment setting, and **the TOE access function** for controlling connection sessions of the authorized administrator.

The end-user identification and authentication process is divided into the initial authentication phase using the ID/password and the token-based authentication phase that accesses the business system using the token issued during the initial authentication procedure.



[Figure 1-1] End-user identification and authentication procedure

The procedure of the initial authentication step is as follows.

When the end-user accesses the business system(A), the SSO agent checks the end-user's authentication status and, if unauthenticated, redirects to the login page of the SSO server. The end-user requests login verification to the SSO server using ID/PW, and the SSO server performs the login verification using the end-user information stored in the DBMS. If the login verification result is valid, the SSO server issues a code[1] to the SSO agent of the business system(A) through the end-user's browser. The SSO agent obtains an authentication token from the SSO server by using the issued code.

The token-based authentication step is performed only after the authentication token has been successfully issued through the initial authentication phase. The end-user requests token-based authentication to the SSO target business system(B) via business system(A) with the issued authentication token in the initial authentication phase. The SSO server issues a code to the SSO agent of business system(B) via the end-user's browser, and the SSO agent receives an authentication token from the SSO server by using the issued code.

[1] code: One-time random number to issue an authentication token

Authentication phase	Operation procedure
Initial authentication	(A) Business system Access – (B) Login request and Login verification – (C) Code issue – (D) Token issue
Token-based authentication	(1) Token-based authentication request– (2) Code request – (3) Code issue – (4) Token issue

[Table 1-4] Operation procedure by authentication phase

- Authentication token issuer: Pass-Ni SSO Server
- Authentication token storage location: Pass-Ni SSO Server, Pass-Ni SSO Agent
- Authentication token validator: Pass-Ni SSO Server

1.3.3 1.3.3(Non-TOE) Hardware/Software/Firmware, Out of TOE required evaluation target

The TOE, in the form of software, is installed in a server or a PC and operated on an operating system (OS) such as Windows or Linux. The hardware / software, which is out of the TOE evaluation target, required for SSO server operation is identified as follows.

Type		Requirements for SSO Server
H/W	CPU	Intel® Xeon^{TM} E5 2.0Ghz or higher
	HDD	Space required for installation of TOE 50GB or higher
	Memory	8GB or higher
	NIC	Ethernet 100/1000 Mbps * 1 port or higher
OS		Red hat Enterprise Linux 8.5 x64
S/W		OpenJDK 11.0.21 Apache Tomcat 9.0.84 MariaDB 10.6.16

[Table 1-5] Hardware/Software Requirements for SSO Server

The hardware / software, which is out of the TOE evaluation target, required for the SSO Agent operation is identified as follows.

Type		Requirements for SSO Agent
H/W	CPU	Intel® Core^{TM} i3 3.6 Ghz or higher
	HDD	Space required for installation of TOE 10GB or higher
	Memory	8GB or higher
	NIC	Ethernet 100/1000 Mbps * 1 port or higher
OS		Red hat Enterprise Linux 8.5 x64 Windows Server 2019 x64
S/W		OpenJDK 11.0.21 Apache Tomcat 9.0.84

[Table 1-6] Hardware/Software Requirements for SSO Agent

The software requirements of the PC used by the authorized administrator to manage the TOE are as follows.

Type	Requirements for administrator's PC
S/W	Google Chrome 110

[Table 1-7] Software Requirements for administrator's PC

The major roles of hardware / software other than the evaluation target required for the TOE operation are as follows.

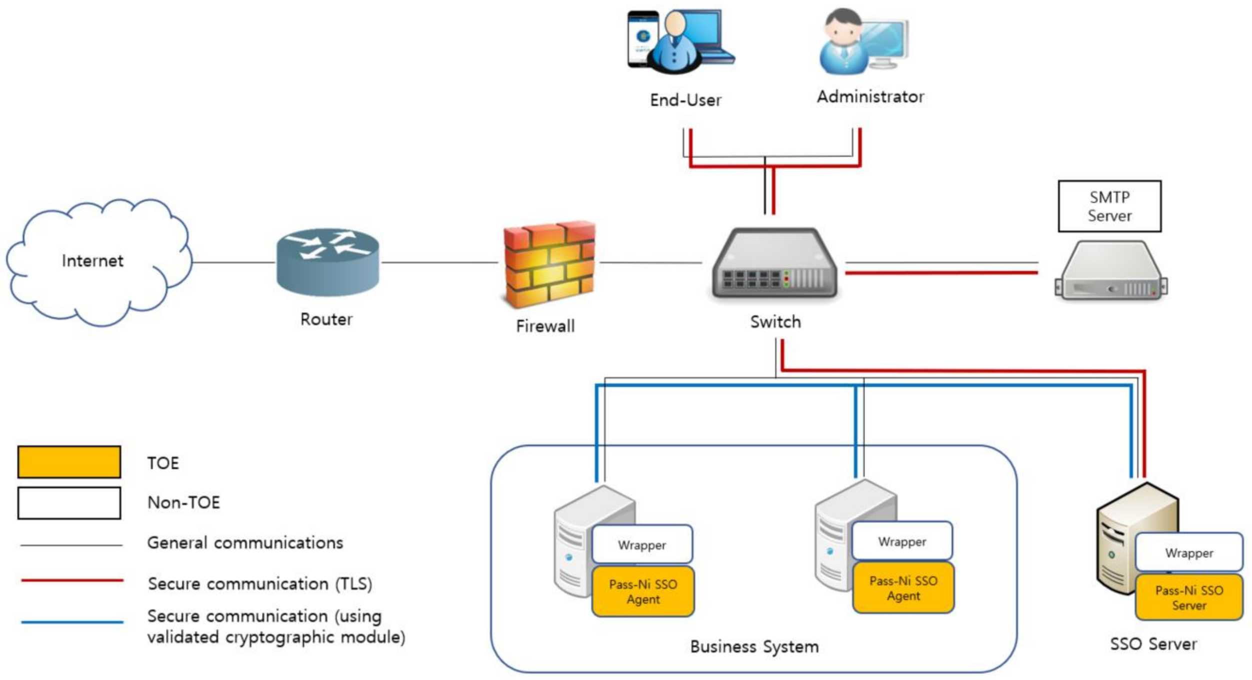
- H/W and OS: Provides operational environment that ensures the reliability and availability of the TOE.
- MariaDB: Records and stores security policies required for the operation of the TOE and security audit data generated during the operation of the TOE
- Apache Tomcat: A web application server to provide security and management functions, providing administrator PC and trusted channel (TLS).
- OpenJDK: The runtime platform for running Java applications
- Google Chrome: The web browser to access TOE security management interface

1.4 TOE Description

This section prescribes the TOE operational environment, the physical scope and the logical scope.

1.4.1 TOE Operational Environment

The operational environment of the TOE is shown in the following figure.



[Figure 1-2] TOE Operational environment

The TOE comprised of the Pass-Ni SSO Server that performs functions such as processing user login, issuing authentication token and managing policy, and the Pass-Ni SSO Agent that is installed in each business system and verifies the validity of the authentication token.

The major roles of the operational services other than the TOE evaluation target required for the TOE operation are as follows.

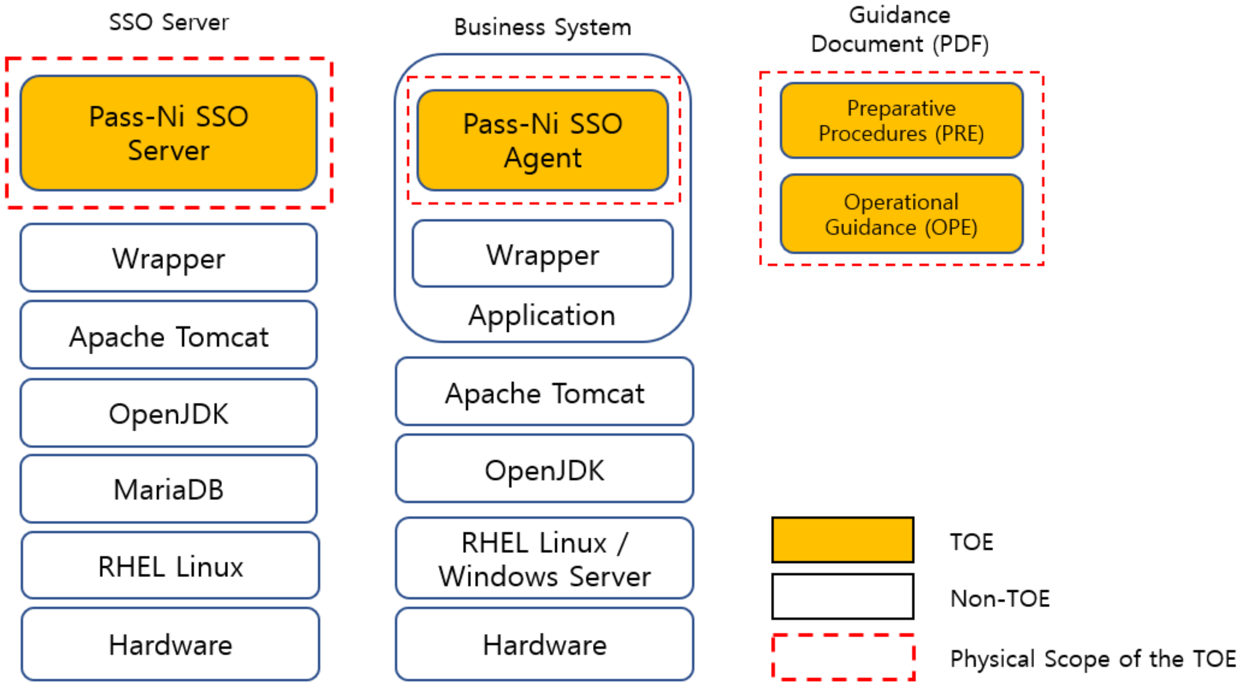
- SMTP server: The mail server that sends an administrator notification mail, such as handling authentication failures and saturation of audit storage

1.4.2 The physical scope of the TOE

The physical scope of the TOE consists of **Pass-Ni SSO Server** which performs functions such as user login processing, authentication token issuance and policy setting, and **Pass-Ni SSO Agent** which is installed in each business system and validates the authentication token through interworking with SSO server. It also includes **preparative procedures** that describe the procedures for secure acceptance and installing the TOE, and **operational guidance** that specify how to use the TOE safely.

The hardware and operating system where the TOE is installed, an administrator PC connect as privileged mode for the TOE security management, the DBMS storing the security policy and audit data, and the Wrappers which may be used to support various types of compatibility with business systems are excluded from the TOE physical scope.

The physical scope of the TOE is shown in the following figure.



[Figure 1-3] Physical Scope of the TOE

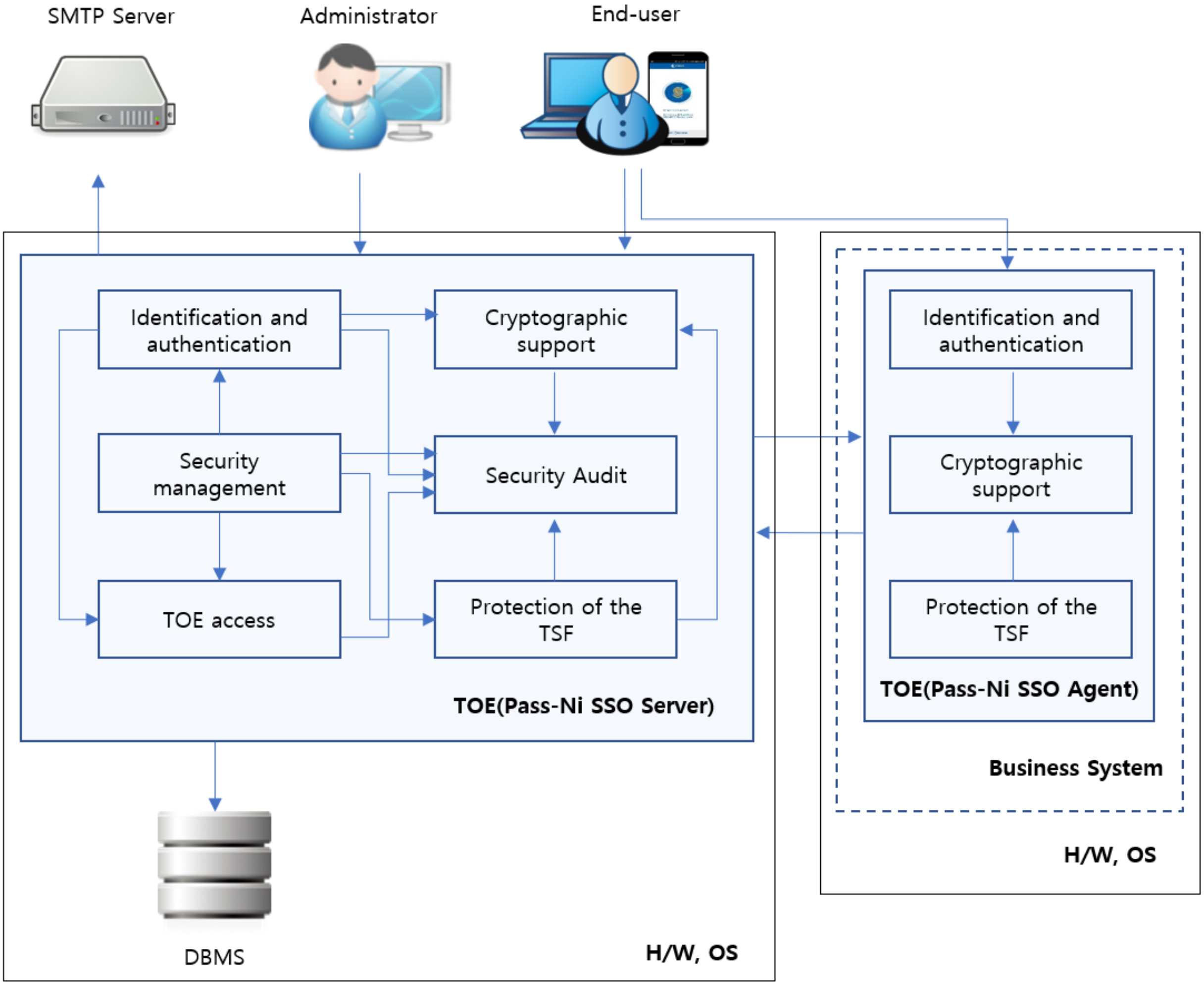
The TOE distributed by the purchaser is distributed as files on the package CD and comprised of the following components.

Item	Identification	Type	Distribution type
TOE	Pass-Ni SSO v5.0 (Detailed version v5.0.002)		CD
TOE Components	Pass-Ni SSO Server v5.0.002 (PassNi-SSO-Server-v5.0.002.zip)	S/W	
	Pass-Ni SSO Agent v5.0.002 (PassNi-SSO-Agent-v5.0.002.zip)	S/W	
Guidance Document	Pass-Ni SSO v5.0 Preparative Procedures V1.0 R3 (PassNi-SSO-v5.0-PRE-V1.0.R3.pdf)	PDF	
	Pass-Ni SSO v5.0 Operational Guidance V1.0 R3 (PassNi-SSO-v5.0-OPE-V1.0.R3.pdf)	PDF	

[Table 1-8] Physical Scope of the TOE

1.4.3 The logical scope of the TOE

The logical scope of the TOE is shown in following figure.



[Figure 1-4] Logical Scope of the TOE

Security audit

The SSO Server generates and stores the audit data for the identification and authentication success / failure of the user, the TOE configuration change history, and the security function execution history. The audit data includes the date and time of the event, the type of the event, the identity of the entity that generates the event, the details of the activity and the results (success / failure), and uses the time information of the TOE installed system to generate accurate time information. The SSO Server allows the authorized administrator to review the audit data and provides the function to perform selectable audit review according to the type of audit data, time, and so on. The SSO Server also provides the function of notifying the authorized administrator through e-mail when the size of the audit trail exceeds the specified limit or when the potential security violation is detected through analysis of the audit data.

Cryptographic support

The SSO Server and the SSO Agent generates a cryptographic key using the target algorithm of the evaluation in the validated cryptographic module whose security and implementation conformance has been verified through the Korean Cryptographic Module Validation Program (KCMVP). It provides the function to securely discard the cryptographic key. The SSO Server and the SSO Agent exchange cryptographic keys through the validated cryptographic module for cryptographic communication between the components and perform functions of symmetric key cryptography, MAC, hash and ECDH cryptographic operation to protect transmitted data and stored data. The SSO Server and the SSO Agent generates a random number using the random number generator of the validated cryptographic module.

Identification and authentication

The SSO server identifies and authenticates administrators using security management functions based on ID / PW. The SSO server identifies and authenticates the end-user based on the ID / PW and issues an authentication token to the authorized end-user. When a end-user requests token-based authentication with an authentication token that has been issued, the SSO Agent verifies the authentication token from the SSO server and performs identification and authentication. The SSO Server will disable the identification and authentication function for the time set by the authorized administrator (default 5 minutes) so that the user can no longer log in if the user fails authentication more than 5 times. During user and administrator identification and authentication process, the TOE ensures that passwords input are masked and prevented from reusing user's authentication information through checking combination rules when generating and changing passwords. The TOE Sever generates an authentication token including a random number and an issuance time (timestamp) to guarantee uniqueness, and securely destroys the authentication token when the session ends or expires. SSO Server and SSO Agent perform mutual authentication through self-implemented authentication protocol.

Security management

The SSO Server provides the security management function that allows the authorized administrator to set and manage security functions, security policies and important data, and rules for creating and changing ID / password. In addition, only for authorized administrator can be accessible, it provides the permission management function that can restrict the functions that can be accessed by administrator and its roles (Top administrator, Monitoring administrator).

Protection of the TSF

The SSO Server and the SSO Agent ensure the confidentiality and integrity of transmitted data and stored TSF data The SSO Server and the SSO Agent performs self-testing periodically during start-up, during normal operation, or at the request of an authorized administrator to ensure the correct operation of each execution module, and to verify the integrity of important data such as execution codes and configurations. The SSO server communicates with the SMTP server through a secure channel and performs an external entity test to confirm that the SMTP server is operating normally when requested by the authorized administrator.

TOE access

Since establishing session of the administrator account from one terminal, the SSO Server terminates the previous session when the access is occurred by the identical account or the identical level administrator from another terminal. Each administrator session or user session is also terminated if the session is not active for a certain period of time (default: 10 minutes) after an administrator or a user login. The SSO server can block the administrator's management access according to the connection IP, whether or not to use, the start date, and the end date.

1.5 Conventions

The notation, formatting and conventions used in this ST are consistent with the Common Criteria for Information Technology Security Evaluation.

The CC allows several operations to be performed for functional requirements: iteration, assignment, selection, and refinement. Each operation is used in this ST.

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

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

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

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

“Application notes” is provided to clarify the intent of requirements, provide the information for the optional items in implementation, and define "Pass/Fail" criteria for a requirement. The application notes are provided with corresponding requirements if necessar.

1.6 Terms and Definitions

The terms used in this ST are the same as those used in the CC and PP, conform to the CC and are not further described in this ST.

JWT (JSON Web Token)

It is composed of three areas (Header, Payload, Signature) in JSON expression format defined by RFC7519 standard

Link System

A business system that is installed with the SSO agent that is the TOE component and works with the SSO server

Link System ID

ID value for uniquely identifying the Link System

Link System SecretCode

A secret value to prove to the SSO server that it is a Link System

SMTP Server

A server that sends e-mail using Simple Mail Transfer Protocol (SMTP)

1.7 Security Target Contents

Chapter 1 Introduction describes the Security Target and TOE reference, TOE overview, TOE description, convention and terms and definitions

Chapter 2 Conformance Claims describes the conformance with the Common Criteria, protection profile and package and presents the conformance rationale and protection profile conformance statement.

Chapter 3 defines the security objectives for the operational environment supported by the operational environment in order to provide the security functionality of the TOE in an accurate manner.

Chapter 4 defines the extended components additionally needed according to the features of Single Sign-On.

Chapter 5 Security Requirements describes security functional requirements and security assurance requirements.

Chapter 6 describes the TOE summary specification.

2. Conformance Claim

This chapter describes how this Security Target complies with the Common Criteria, Protection Profile, and Package.

2.1 CC Conformance Claim

This Security Target conforms to the following Common Criteria.

Common Criteria		Common Criteria for Information Technology Security Evaluation V3.1R5 - Common Criteria Part 1: Introduction and General Model V3.1r5, (CCMB-2017-04-001, 2017. 4) - Common Criteria Part 2: Security Functional Components V3.1r5, (CCMB-2017-04-002, 2017.4) - Common Criteria Part 3: Security Assurance Components V3.1r5, (CCMB-2017-04-003, 2017.4)
Conformance Claim	Part2 Security Functional Requriements	Extended: FCS_RBG.1, FIA_IMA.1, FIA_SOS.3, FMT_PWD.1, FPT_PST.1, FTA_SSL.5
	Part 3 Security Assurance Requirements	Conformant
	Package	Augmented: EAL1 augmented(ATE_FUN.1)

2.2 PP Conformance Claim

This Security Target conforms to the following Protection Profile.

- Protection Profile
  - National Protection Profile for Single Sign-On V1.1 (KECS-PP-0822a-2017, 2019. 12. 11)
- PP Conformance Type

- "Strict PP conformance"

2.3 Package Conformance Claim

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

- Assurance package: EAL1 augmented(ATE_FUN.1)

2.4 Conformance Claim Rationale

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

The rationale for the PP conformance claim in this Security Target is as follow.

Item	PP	ST	Conformance Rationale
Seucirty objectives	OE.PHYSIAL_CONTROL	OE.PHYSIAL_CONTROL	Same as PP
	OE.TRUSTED_ADMIN	OE.TRUSTED_ADMIN	
	OE.LOG_BACKUP	OE.LOG_BACKUP	
	OE.OPERATION_SYSTEM_REINFORCEMENT	OE.OPERATION_SYSTEM_REINFORCEMENT	
	OE.SECURE_DEVELOPMENT	OE.SECURE_DEVELOPMENT	
	OE.AUTHENTICATION_SYSTEM_SECURITY	-	Exclude from PP - In the initial authentication stage of the TOE, identification and authentication functions of endusers are not upported by external authentication systems nd therefore, the security goal of 'OE.AUTHENTICATION_SYSTEM_SECURITY' does not

			correspond.
	-	OE.TRUSTED_TIMESTAMP	More restrictive than PP - This ST is further defined in this ST for security objectives that must be addressed by the technical / procedural means supported by the operational environment to provide security functionality. Therefore, this ST is more restrictive than PP because it defines 'the TOE operating environment to deal with these additional security objectives'
	-	OE.TRUSTED_SMTP	
	-	OE.TRUSTED_AUDIT_STORAGE	
	-	OE.SECURE_CHANNEL	

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

This ST conforms to the security objectives for all operating environments specified in the PP. The following are security objectives to be addressed by the technical / procedural means supported by the operational environment so that the TOE can accurately provide security functionality.

- **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 checks 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.SECURE_CHANNEL**

The Web Application Server (WAS) that interfaces with the TOE provides a secure and trusted channel so that all information transmitted when the user accesses through the web browser should be securely protected.

- **OE.TRUSTED_TIMESTAMP**

The TOE should be used with reliable time information provided by TOE operating environment.

- **OE.TRUSTED_SMTP**

The authorized administrator of the TOE shall set secure and reliable SMTP server information so that the TOE can send mail to the secure path when installing the TOE.

- **OE.TRUSTED_AUDIT_STORAGE**

The audit storage associated with the TOE must ensure that it maintains secure and trusted operations.

4. Extended components definition

4.1 Cryptographic support (FCS, 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 Random bit generation

1

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 and 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 TOE Internal mutual authentication
1

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

Management: FIA_IMA.1

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

FIA_SOS Specification of Secrets
1
2
3

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*].

Application notes

○ This SFR can be applied to the user's token.

4.3 Security Management

4.3.1 ID and password

Family Behaviour
This family defines the capability that is required to control ID and password management used in the TOE, and set or modify ID and/or password by authorized users.

Component leveling

FMT_PWD ID and password
1

FMT_PWD.1 ID and password management, requires that the TSF provides the 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*].

Application notes

○ If the TOE does not provide the capability for managing the ID and password combination rules by authorized roles, etc., 'None.' may be specified in assignment operations of FMT_PWD.1.1, FMT_PWD.1.2.

○ The ID and password combination rules that can be set by authorized roles may include minimum and maximum length setting, mixing rule setting involving English upper case/lower case/number/special characters, etc.

4.4 Protection of the TSF

4.4.1 Protection of stored TSF data

Family Behaviour

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

Component leveling

FPT_PST Protection of stored TSF data — 1

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*].

Application notes

○ Containers controlled by the TSF mean storage in the TOE or external entities (DBMS, etc.) that interact with the TOE.

○ Examples of TSF data to be protected as follows:

- User password, cryptographic key (pre-shared key, symmetric key, private key, etc), TOE configuration values (security policy, environment setting, configuration parameters), audit data, etc.

○ The TSF data can be encrypted and stored to be protected from the unauthorized disclosure or modification.

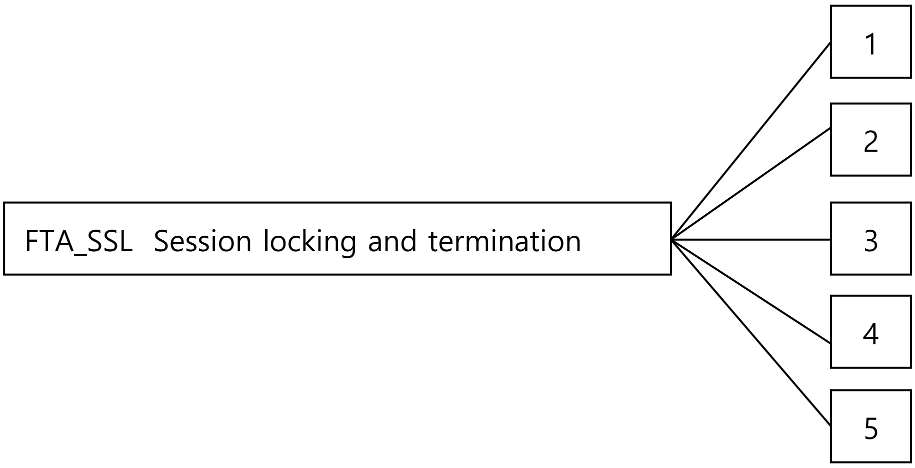
4.5 TOE Access

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].*

Application notes

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

5. Security requirements

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

The subjects, objects, operations, and security attributes used in the security functional requirements of the TOE are as follows.

Subjects	Subjects security attributes	Objects	Objects security attributes	Operations
SSO Server	Signature key (RSA PrivateKey)	Cryptographic key	-	generation, distribution, operation, destruction
		Audit data	date and time of the event, type of event, subject identity, the outcome (success or failure) of the event.	generation
		Authentication token	Issuser, Subject(user), IP Address, Audience, Time of issue, Time of expiration, Integrity verification code	Issuance, Verification, Destruction
SSO Agent	Secret Key (HMAC Key)	Cryptographic key	-	generation, distribution, operation, destruction
		Authentication token	Issuser, Subject(user), IP Address, Agent-ID, Time of issue, Time of expiration, Integrity verification code	Issuance request, verification request, destruction request
Authorized Administrator	User ID, Password, IP address, Roles	Policy data	UserID policy, Password policy, User policy, Administrator policy,	query, modify

			Audit violation policy	
		Business System infomation	IP, Secret	generation, query, modify, delete
		End-User infomation	User ID	generation, query, delete
			Password	generation, reset
		Administrotor infomation	User ID	generation, query, delete
			Password	generation, reset, change
			IP address	generation, query, modify, delete
			Role	change
		Audit data	Date and time of the event, Type of event, Subject identity, The outcome (success or failure) of the event.	query, search
Authorized End-User	UserId, Password	End-User infomation	User ID	Identification and authentication
			Password	Identification and authentication, change
		Authentication token	Issuser, Subject(user), IP Address, Agent-ID, Time of issue, Time of expiration, Integrity verification code	Identification and authentication

The external entities required for the operation of the TOE are as follows.

- **SMTP**: The mail server that sends an administrator notification mail, such as handling authentication failures and saturation of audit storage

5.1 Security functional requirements

The following table summarizes the security functional requirements used in the ST

Security Functional Class	Security Functional Component		Remarks
Security Audit (FAU)	FAU_ARP.1	Security alarms	
	FAU_GEN.1	Audit data generation	
	FAU_SAA.1	Potential violation analysis	
	FAU_SAR.1	Audit review	
	FAU_SAR.3	Selectable audit review	
	FAU_STG.3	Action in case of possible audit data loss	
	FAU_STG.4	Prevention of audit data loss	
Cryptographic Support (FCS)	FCS_CKM.1	Cryptographic key generation	
	FCS_CKM.2	Cryptographic key distribution	
	FCS_CKM.4	Cryptographic key destruction	
	FCS_COP.1(1)	Cryptographic operation (Symmetric key cryptographic operation)	
	FCS_COP.1(2)	Cryptographic operation (MAC)	
	FCS_COP.1(3)	Cryptographic operation (Hash)	
	FCS_COP.1(4)	Cryptographic operation (ECDH)	
	FCS_COP.1(5)	Cryptographic operation (Digital signature)	
	FCS_RBG.1(Extended)	Random bit generation	
Identification and Authentication (FIA)	FIA_AFL.1(1)	Authentication failure handling (End-user)	
	FIA_AFL.1(2)	Authentication failure handling (Administrator)	
	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	User authentication before any action	
	FIA_UAU.4	Single-use authentication mechanisms	
	FIA_UAU.7	Protected authentication feedback	
	FIA_UID.2	User identification before any action	
Security Management (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	
Protection of the TSF (FPT)	FPT_ITT.1	Basic internal TSF data transfer protection	
	FPT_PST.1(Extended)	Basic protection of stored TSF data	
	FPT_TST.1	TSF testing	
TOE Access (FTA)	FTA_MCS.2	Per user attribute Limitation on multiple concurrent sessions	
	FTA_SSL.5(Extended)	Management of TSF-initiated sessions	
	FTA_TSE.1	TOE session establishment	

[Table 5-1] Security functional requirements (SFR)

5.1.1 Security audit (FAU)

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 [Sending mail to the e-mail address specified by the authorized administrator] upon detection of a potential security violation.

5.1.1.2 FAU_GEN.1 Audit data generation

Hierarchical to No other components.

Dependencies FPT_STM.1 Reliable time stamps

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

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

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

c) [Refer to the “auditable events” in **[Table 5-2]** Audit events**,** [N/A]]

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 5-2]** Audit events, [N/A]].

Security functional component	Auditable event	Additional audit record
FAU_ARP.1	Actions taken due to potential security violations	
FAU_SAA.1	Enabling and disabling of any of the analysis mechanisms, Automated responses performed by the tool	
FAU_STG.3	Actions taken due to exceeding of a threshold	
FAU_STG.4	Actions taken due to the audit storage failure	
FCS_CKM.1	Success and failure of the activity	
FCS_CKM.2	Success and failure of the activity (only applying to key distribution related to the TSF data encryption/decryption)	
FCS_CKM.4	Success and failure of the activity (only applying to key destruction related to the TSF data encryption/decryption)	
FCS_COP.1	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_IMA.1(Extended)	Success and failure of mutual authentication	
FIA_SOS.2	Rejection by the TSF of any tested secret	
FIA_SOS.3(Extended)	Success and failure of the activity(applicable to the destruction of SSO token only)	
FIA_UAU.2	All use of the authentication mechanism	
FIA_UAU.4	Attempts to reuse authentication data	
FIA_UID.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	

[Table 5-2] Audit events

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 [
An auditable event of authentication failure in FIA_UAU.2,
An auditable event of integrity violation in FPT_TST.1,
Self-test failure of the validated cryptographic module,
The audit tail storage threshold exceeded,
The audit tail storage is full
] known to indicate a potential security violation;

b) [N/A]

5.1.1.4 FAU_SAR.1 Audit review

Hierarchical to No other components.

Dependencies FAU_GEN.1 Audit data generation

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

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

5.1.1.5 FAU_SAR.3 Selectable audit review

Hierarchical to No other components

Dependencies FAU_SAR.1 Audit review

FAU_SAR.3.1 The TSF shall provide the ability to apply [the following methods of selection and ordering] of audit data based on [criteria with the following logical relations].

Audit type	Criteria with Logical Relations	Allowable Ability
System Audit	- Search period (start date, end date) AND - Status(success, failure) AND - Type(Server start, Server stop) AND - IP address OR Result code	Search, Sort, View details
Business System Audit	- Search period (start date, end date) AND - Status(success, failure) AND - Type(Agent start, Agent stop) AND - Biz system ID OR Biz system name	
Administrator access	- Search period (start date, end date) AND - Status(success, failure) AND - User ID OR Aame OR IP address	Search, Sort, View details
Administrator action		
End-user access		
End-user action		

[Table 5-3] Criteria with logical relations by audit type

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

Hierarchical to No other components

Dependencies FAU_GTG.1 Protected audit trail storage

FAU_STG.3.1 The TSF shall [Notification to the authorized administrator, [N/A] if the audit trail exceeds [80% of database's volume capacity (fixed value)]].

5.1.1.7 FAU_STG.4 Prevention of audit data loss

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

Dependencies FAU_GTG.1 Protected audit trail storage

FAU_STG.4.1 The TSF shall "*overwrite the oldest stored audit records*" and [Sending mail to the e-mail address specified by the authorized administrator] if the audit trail is full.

5.1.2 Cryptographic support (FCS)

5.1.2.1 FCS_CKM.1 Cryptographic key generation

Hierarchical to No other components

Dependencies [FCS_CKM.2 Cryptographic key distribution, or
FCS_COP.1 Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction

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

Item	Cryptographic Key generation Algorithm	Detail	cryptographic Key Sizes (bits)	List of Standards
KEK	PBKDF2	HMAC-SHA256 Salt: 128bits Iteration: 1000	256	TTAK.KO-12.0274

Master Key	Hash_DRBG	Hash: SHA-256	512 (256,256)	ISO/IEC 18031
Secret Key	Hash_DRBG	Hash: SHA-256	256	ISO/IEC 18031
Elliptic curve key pair	Hash_DRBG	Hash: SHA-256	256	ISO/IEC 18031
Shared Key	ECDH	Hash: SHA-256 Curve: P-256	512 (256,256)	ISO/IEC 11770-3
Digital signature key pair	Hash_DRBG	Hash: SHA-256	2048 (public key)	ISO/IEC 18031

[Table 5-4] Cryptographic key generation algorithm

Application Notes

○ The cryptographic algorithm and cryptographic key sizes shall meet the cryptographic complexity of 112 bits or more.

○ Generating a cryptographic key by deriving it from the password is not allowed, except the key encryption key (KEK).

5.1.2.2 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 [ECDH (Elliptic Curve Diffie-Hellman) elliptic-curve based key establishment mechanisms (Curve: P-256)] that meets the following: [ISO/IEC 11770-3].

5.1.2.3 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 [Overwrite 3 times with 0] that meets the following: [none].

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

Hierarchical to No other components

Dependencies [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1(1) The TSF shall perform ["list of cryptographic operations" in the following table] in accordance with a specified cryptographic algorithm ["cryptographic algorithm" in the following table] and cryptographic key sizes ["Key sizes" in the following table] that meet the following: ["list of standards" in the following table]

List of standards	Cryptographic alogrithm	Key sizes(bits)	List of cryptographic operations
KS X 1213-1	ARIA-CBC	256 bits	- TSF data - Transfer data between SSO Server and Agent - Authentication token
TTAS.KO-12.0004/R1	SEED-CBC	128 bits	- Private key of the TOE

[Table 5-5] List of cryptographic operation (Symmetric key)

5.1.2.5 FCS_COP.1(2) Cryptographic operation (MAC)

Hierarchical to No other components

Dependencies [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1(2) The TSF shall perform ["list of cryptographic operations" in the following table] in accordance with a specified cryptographic algorithm

["cryptographic algorithm" in the following table] and cryptographic key sizes ["key sizes" in the following table] that meet the following: ["list of standards" in the following table]

List of standards	Cryptographic alogrithm	Key sizes(bits)	List of cryptographic operations
ISO/IEC 9797-2	HMAC-SHA256	256 bits	- TOE Internal mutual authentication - Integrity verification of transfer data between SSO Server and Agent - Integrity verification of the authentication token

[Table 5-6] List of cryptographic operation (MAC)

5.1.2.6 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(3) The TSF shall perform ["list of cryptographic operations" in the following table] in accordance with a specified cryptographic algorithm ["cryptographic algorithm" in the following table] and cryptographic key sizes ["cryptographic complexity" in the following table] that meet the following: ["list of standards" in the following table]

List of standards	Cryptographic alogrithm	cryptographic complexity[2] (bits)	List of cryptographic operations
ISO/IEC 10118-3	SHA-256	128	- End-user/administrator's password - TOE's Integrity verification code

[Table 5-7] List of cryptographic operation (Hash)

5.1.2.7 FCS_COP.1(4) Cryptographic operation (ECDH)

Hierarchical to No other components

Dependencies [FDP_ITC.1 Import of user data without security attributes, or

[2] The hash cryptographic operation does not use a cryptographic key, so it is replaced by a cryptographic complexity(security strength)

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(4) The TSF shall perform [“list of cryptographic operations” in the following table] in accordance with a specified cryptographic algorithm [“cryptographic algorithm” in the following table] and cryptographic key sizes [“key sizes” in the following table] that meet the following: [“list of standards” in the following table]

List of standards	Cryptographic alogrithm	Key sizes(bits)	List of cryptographic operations
ISO/IEC 11770-3	ECDH Curve: P-256	256 bits	- Distributing cryptographic keys between TOE components

[Table 5-8] List of cryptographic operation (ECDH)

5.1.2.8 FCS_COP.1(5) Cryptographic operation (Digital Signature)

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(5) The TSF shall perform [“list of cryptographic operations” in the following table] in accordance with a specified cryptographic algorithm [“cryptographic algorithm” in the following table] and cryptographic key sizes [“key sizes” in the following table] that meet the following: [“list of standards” in the following table]

List of standards	Cryptographic alogrithm	Key sizes(bits)	List of cryptographic operations
KS X ISO/IEC 14888-2	RSA-PSS Hash: SHA256	2048 bits	- Digital Signature and verification between TOE components

[Table 5-9] List of cryptographic operation (Digital Signature)

5.1.2.9 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 [ISO/IEC 18031 Hash_DRBG(SHA-256)].

5.1.3 Identification and authentication (FIA)

5.1.3.1 FIA_AFL.1(1) Authentication failure handling (End-user)

Hierarchical to No other components

Dependencies FIA_UAU.1 Timing of authentication

FIA_AFL.1.1(1) The TSF shall detect when *an administrator configurable positive integer within [3 ~ 10 (default value 5)]* unsuccessful authentication attempts occur related to [authentication attempt of end-user].

FIA_AFL.1.2(1) When the defined number of unsuccessful authentication attempts has been *met* the TSF shall [the following list of actions

a) Disable the user identification and authentication function during a positive number of minutes (5 to 60) configurable by the administrator (temporary blocking, default value 5 minutes) or

b) Disable (block) the identification and authentication functions until the administrator unlocks disabled identification and authentication functions for the user

].

5.1.3.2 FIA_AFL.1(2) Authentication failure handling (Administrator)

Hierarchical to No other components

Dependencies FIA_UAU.1 Timing of authentication

FIA_AFL.1.1(2) The TSF shall detect when *an administrator configurable positive integer within [3 ~ 10 (default value 5)]* unsuccessful authentication attempts occur related to [authentication attempt of administrator].

FIA_AFL.1.2(2) When the defined number of unsuccessful authentication attempts has been *met* the TSF shall [the following list of actions

a) Disable the user identification and authentication function during a positive number of minutes (5 to 60) configurable by the administrator (temporary blocking, default value 5 minutes) or

b) Disable (block) the identification and authentication functions until the administrator unlocks the disabled identification and authentication functions for the user].

5.1.3.3 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 [Pass-Ni SSO Server, Pass-Ni SSO Agent] using the [self-implemented authentication protocol] that meets the following [N/A]

5.1.3.4 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 [Administrator-defined permission criteria in FMT_PWD.1].

Application notes

○ The information that shall meet password complexity requirements can be data as the following

- administrator's password, end-user's password

5.1.3.5 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 [the following acceptable standard

a) The subject of token generation and authentication is the SSO server.

b) The authentication token contains components as [Table 5-10].

c) The header and data contained in the authentication token must ensure integrity.

d) The data contained in the authentication token must be encrypted with a validated cryptographic module to provide confidentiality.

e) Among the authentication token components, 'subject (user)' information must ensure uniqueness and generated using a random bit generator of a validated cryptographic module.

]

FIA_SOS.2.2 TSF shall be able to enforce the use of TSF-generated **authentication token** for [End-user identification and authentication].

Type	Composition	Description	subject of generation
Header	Algorithm	Signature Alogorithm (default: HMAC-SHA256)	SSO Server
Payload Header	Issuser	SSO server ID that issued the authentication token	
	Subject(user)	Session ID to identify the user	
	IP	User access ip address	
	Browser	User access browser identification	
	Audience	SSO Agent ID for which authentication token is issued	
	TIme of issue	Time that the authentication token was issued	
	Time of expiration	Expiration time of authentication token	
Payload	Integrity verification code	Header and data integrity verification code with algorithm defined in header	

[Table 5-10] Structure of authentication token

5.1.3.6 FIA_SOS.3 Destruction of secrets (Extended)

Hierarchical to No other components

Dependencies FIA_SOS.2 TSF Generation of secrets.

FIA_SOS.3.1 The TSF shall destroy **authentication tokens** in accordance with a specified **authentication token** destruction method [Overwrite 3 times with 0] that meets the following: [N/A]

5.1.3.7 FIA_UAU.2 User authentication before any action

Hierarchical to FIA_UAU.1 Timing of Authentication

Dependencies FIA_UID.1 Timing of identification

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

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 [
the following identified authentication mechanism(s)
a) Password based authentication (end-user, administrator)
b) Authentication token based authentication (end-user)
].

Application notes

○ This SFR defines the requirements for the authentication data and token of the authorized administrator and the authorized end-user.

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 [the following list of feedback
a) Passwords being entered are masked (password masking with "● ● ● ● ●") during administrator / user password creation, change and administrator / user authentication
b) When administrator identification and authentication fails, the following message
- User id or password is incorrect.
c) When end-user identification and authentication fails, the following

message

- User id or password is incorrect.

] to the user while the authentication is in progress.

5.1.3.10 FIA_UID.2 User identification before any action

Hierarchical to FIA_UID.1 Timing of identification

Dependencies No dependencies

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

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 [list of functions in the following table] to [the authorized administrator].

List of functions	**Management actions**					**Administrator**
	determine the behavior	**disable**	**enable**	**modify the behaviour of**	**remarks**	
Management of id configuration rules	-	○	○	○		Top administrator
Management of password configuration rules	-	○	○	○		
Management of actions to be taken in the event of an	-	○	○	○		

authentication failure(end-user)						
Management of actions to be taken in the event of an authentication failure(administrator)	-	-	-	○		
Actions to be taken when authentication is attempted with password change period exceeded (end-user)	-	○	○	○		
Actions to be taken when authentication is attempted with password change period exceeded (administrator)	-	-	-	○		
Actions to be taken in event of inactivity timeout(end-user)	-	○	○	○	Administrators have fixed values	
Management of actions to be taken in the event of a potential violation	-	○	○	-	Whether to send mail	
Management of server self-tests	-	-	-	○	Execution time	
Management of agent self-tests	-	-	-	○	Execution time	

[Table 5-11] List of security management functions

Application notes

○ "Management action" to which a refinement operation is applied includes the ability to determine the behavior, disable, enable, modify the behavior of some functions in the TSF. This requirement shall be applied to the management access(SSH, HTTPS, etc.) supported by the TOE

○ The action that adds, deletes or modifies conditions or rules capable of determining the security functions behavior is included in the management of security functions behaviors. And, the action that adds, deletes or modifies behaviors taken by the TSF according to the corresponding conditions and rules is also included in the management of security functions behaviors. In addition, the action of selecting mechanism, protocol, etc., when there are variously provided to support the same purpose, is included in the management of security functions behavior because it corresponds to the modification of behavior.

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 [list of TSF data in the following table] to [the authorized administrator in the following table].

TSF data	Ability						Administrator
	change default	query	modify	delete	clear	other operation	
Audit data	-	○	-	-	-	Audit view, statistics	top administrator, monitoring administrotor
User access status	-	○	-	-	-	Force logout	top administrator
User information	-	○	○	○	-	Password reset, Unblock	
ID policy	-	○	○	-	-	-	
Password policy	-	○	○	-	-	-	
End-user policy	-	○	○	-	-	-	
Administrator policy	-	○	○	-	-	allowed ip number	
Audit violation policy	-	○	○		-		

Administrator information	-	○	○	○	-	Password reset, Unblock	
Integrity verification data	-	○	-	-	-	Server/Agent Self-Test	
Business system information	-	○	○	○	-	-	
Group of Business system information	-	○	○	○	-	-	
BATCH information	-	○	○	○	-		
Code-defined	-	○	○	-	-	-	
End-user's own Password	-	-	○	-	-	-	End-user

[Table 5-12] List of TSF data

Application notes

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

○ Among the management functions of TSF data specified in above Table, 'Query' function can be performed by all authorized administrators, and 'Modify' and 'Delete' functions can be performed only by the administrator specified in the above table

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 [Creating and changing administrator/end-user passwords] to [the authorized administrator].

1. [

a) Password combination rule: Two or more combinations of lowercase / uppercase / numeric / special characters (default: Three combinations of lowercase letters, numbers, and special characters)

b) Password length: between 6-99 characters (default: minimum 9 characters, maximum 20 characters)

]

2. [

a) other management such as management of special characters unusable for password: None (32 special characters that can be input by the following keyboard {
'`', '~', '!', '@', '#', '$', '%', '^', '&', '*',
'(', ')', '_', '+', '[', ']', '{', '}', ';', ''',
':', '"', ',', '.', '/', '<', '>', '?', '-', '=',
'_', '+' })

b) Number of repetitions of the same character: Limit more than the number set by the authorized administrator (default: 3)

c) Number of consecutive character repetitions: Limit more than the number set by the authorized administrator (default: 3)

d) Number of consecutive characters in the keyboard layout: Limit more than the number set by the authorized administrator (default: 4)

e) Recent password change: limited to the number set by the authorized administrator (default: 1)

f) Password change period: The period set by the authorized administrator (1 day ~ 365 days), change password when it is exceeded (default: 60 days)

]

FMT_PWD.1.2 The TSF shall restrict the ability to manage the ID of [Creating administrator/end-user ID] to [the authorized administrator].

1. [

a) ID combination rule: combinations of lowercase / uppercase / numeric / special characters (default: lowercase letters)

b) ID length: between 5-20 characters (default: minimum 6 characters, maximum 20 characters)

]

2. [

a) Special characters for ID: Only allowed special characters {minus (-), underline (_), point(.)}

]

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

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: [

a) List of security functions specified in FMT_MOF.1

b) Management functions of TSF data specified in FTP_MTD.1

c) Management functions of ID/Password specified in FMT_PWD.1

].

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 [

a) top administrator

b) monitoring administrator

].

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 [TSF data in the following table] stored in the repository, which is controlled by the TSF, from unauthorized *exposure and modification*.

Storage	TSF data	Protection method
DBMS	User's(end-user/administrator) password	Access control, Encryption(Hash)
	Among the authentication token components, 'subject (user)'	Access control, Encryption(Symmetric key)
	Server-Agent Shared key(Symmetric key)	Access control, Encryption(Symmetric key)
	Server-Agent Secret key(MAC key)	Access control, Encryption(Symmetric key)
	TOE configuration value (ID / PW generation rule, Authentication failure handling policy)	Access control, Encryption(Symmetric key)
Filesytem	DBMS account's password	Encryption(Symmetric key)
	SMTP account's password	Encryption(Symmetric key)
	TOE's configuration value (environment configuration parameter)	Encryption(Symmetric key)

[Table 5-13] Protected TSF data

5.1.5.3 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 [*Audit review, Cryptographic support, TSF Generation of secrets, Protection of stored TSF data*].

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

FPT_TST.1.3 The TSF shall provide **authorized administrators** 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 [the number of maximum concurrent sessions as 1 for administrator management access sessions, the number of maximum concurrent sessions as 1 for end-user access sessions, rules for the number of maximum concurrent sessions determined as follow]

{

a) If you log in again with the same account or same privilege, terminate previous connection

b) Monitoring administrator allows duplicate login with other administrator accounts

}

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

Application notes

○ A session is presented in FMT_MCS.2 is 'user access', the number of sessions shall be 'the number of user accesses.'

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 administrator/end-user inactivity

a) administrator: fixed value(10 minutes)

b) end-user: Time set by the authorized administrator(1 ~ 60minutes or not used, default value: 10 minutes)

].

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 **administrator's management access** session establishment based on [connection IP, *[whether or not to use, start date, end date]*].

Application notes

○ The management access session of administrator shall be allowed only from the terminal with designated IP address for management access.

5.2 Security Assurance Requirements

Security assurance requirements of this ST are composed of assurance components in Common Criteria (CC V3.1) Part 3 and the evaluation assurance level is EAL1+(ATE_FUN.1).

The table below summarizes assurance components.

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

[Table 5-14] Assurance Component Summary

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.

5.2.3.1 **AGD_PRE.1 Preprative 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 Labeling of the TOE

Dependencies ALC_CMS.1 TOE CM coverage

Developer action elements

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

Content and presentation elements

ALC_CMC.1.1C	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 - conformane

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

This section sets out the rationale for the dependency on the security functional and warranty requirements to demonstrate that the security requirements described in this ST are appropriate to satisfy the dependency.

5.3.1 Dependency rationale of security functional requirements

The following table shows dependency of security functional requirement.

No.	Security functional requirements	Dependency	Reference No.
1.	FAU_ARP.1	FAU_SAA.1	3
2.	FAU_GEN.1	FPT_STM.1	OE.TRUSTED_TIMESTAMP
3.	FAU_SAA.1	FAU_GEN.1	2
4.	FAU_SAR.1	FAU_GEN.1	2
5.	FAU_SAR.3	FAU_SAR.1	4
6.	FAU_STG.3	FAU_STG.1	OE.TRUSTED_AUDIT_STORAGE
7.	FAU_STG.4	FAU_STG.1	OE.TRUSTED_AUDIT_STORAGE
8.	FCS_CKM.1	[FCS_CKM.2 or FCS_COP.1]	9, 11, 12, 13, 14
		FCS_CKM.4	10
9.	FCS_CKM.2	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	8
		FCS_CKM.4	10
10.	FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	8
11.	FCS_COP.1(1)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	8
		FCS_CKM.4	10
12.	FCS_COP.1(2)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	8
		FCS_CKM.4	10
13.	FCS_COP.1(3)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	8
		FCS_CKM.4	10
14.	FCS_COP.1(4)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	8
		FCS_CKM.4	10
15.	FCS_COP.1(5)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	8
		FCS_CKM.4	10
16.	FCS_RBG.1	-	-

17.	FIA_AFL.1(1)	FIA_UAU.1	23
18.	FIA_AFL.1(2)	FIA_UAU.1	23
19.	FIA_IMA.1	-	-
20.	FIA_SOS.1	-	-
21.	FIA_SOS.2	-	-
22.	FIA_SOS.3	FIA_SOS.2	21
23.	FIA_UAU.2	FIA_UID.1	26
24.	FIA_UAU.4	-	-
25.	FIA_UAU.7	FIA_UAU.1	23
26.	FIA_UID.2	-	-
27.	FMT_MOF.1	FMT_SMF.1 FMT_SMR.1	30 31
28.	FMT_MTD.1	FMT_SMF.1 FMT_SMR.1	30 31
29.	FMT_PWD.1	FMT_SMF.1 FMT_SMR.1	30 31
30.	FMT_SMF.1	-	-
31.	FMT_SMR.1	FIA_UID.1	26
32.	FPT_ITT.1	-	-
33.	FPT_PST.1	-	-
34.	FPT_TST.1	-	-
35.	FTA_MCS.2	FIA_UID.1	26
36.	FTA_SSL.5	FIA_UAU.1 or No dependencies	23
37.	FTA_TSE.1	-	-

[Table 5-15] Rationale for the dependency of the security functional requirements

FAU_GEN.1 has a dependency on FPT_STM.1. However, reliable time stamps provided by the security objective OE.TRUSTED_TIMESTAMP for the operational environment of this ST are used, thereby satisfying the dependency

FAU_STG.3 and FAU_STG.4 have a dependency on FAU_STG.1. However, it is protected from unauthorized deletion or modification in accordance with the security objective OE.TRUSTED_AUDIT_STORAGE for the operational environment of this ST, thereby satisfying the dependency.

FIA_AFL.1(1) and FIA_AFL.1(2) has a dependency on FIA_UAU.1, which is satisfied by

FIA_UAU.2 hierarchical to FIA_UAU.1.

FIA_UAU.2 has a dependency on FIA_UID.1, which is satisfied by FIA_UID.2 hierarchical to FIA_UID.1.

FIA_UAU.7 has a dependency on FIA_UAU.1, which is satisfied by FIA_UAU.2 hierarchical to FIA_UAU.1.

FMT_SMR.1 has a dependency on FIA_UID.1, which is satisfied by FIA_UID.2(1) and FIA_UID.2(2) hierarchical to FIA_UID.1.

FTA_MCS.2 has a dependency on FIA_UID.1, which is satisfied by FIA_UID.2 hierarchical to FIA_UID.1.

FTA_SSL.5 has a dependency on FIA_UAU.1, which is satisfied by FIA_UAU.2 hierarchical to FIA_UAU.1.

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 Summary Specification

This chapter specifies the security functionality of the TOE that satisfies the security functional requirements (SFR).

6.1 TOE Security functionality

TOE security functionalities can be roughly divided into "Security audit, Cryptographic support, Identification and authentication, Security management, Protection of the TSF protection, TOE access.” This section describes how the TOE satisfies the “security functional requirements” specified in the previous section.

6.1.1 Security audit (FAU)

Audit data generation

FAU_GEN.1 TOE manages all auditable events (as defined in FAU_GEN.1) that occur during the operation through the repository (DBMS) of the TOE operation environment for purpose of storing audit data.
The audit data generated by TOE describes the date and time of the event, type of event, the identity of the subject, and the result of the event (success or failure).

Potential violation analysis and Security alarms

FAU_ARP.1
FAU_SAA.1 When TOE detects potential security violation events during the check auditable events, it notifies the authorized administrator via email.

TOE detects the following audit events as potential security violations as defined by the authorized administrator using the TOE management tool(audit violation policy management).

- When the number of failed authentication attempts by user and administrator exceeds a set limit
- When the failure occurs in product self-test(product integrity checks, cryptographic module self-test)
- When the audit storage threshold is exceeded

- When audit storage is reaching full capacity

Audit review

FAU_SAR.1
FAU_SAR.3

TOE provides authorized administrators with the functionality to review all of TOE's audit data, distinguishing it according to types and selection criteria based on these types.
TOE offers the functionality of search and sorting, and detailed viewing of audit data to make it suitable for administrator interpretation.

Detail information about the [Audit Review] function of TOE is specified in Section **6.1.4.5 Audit view and statistics**.

Counteract and prevention of audit data loss

FAU_STG.3
FAU_STG.4

TOE provides functionalities of responding to and preventing the loss of audit data. To prevent losses of audit data, due to insufficient storage capacity, it regularly checks the usage of the storage. TOE sends an email to the email address designated by the authorized administrator when the usage space of the audit trail storage exceeds a specified limit (80% of the database volume capacity).

When the audit trail storage is saturated (meaning the usage space of the audit trail storage reaches 95% of the database volume capacity), TOE sends an email to the specified email address of the authorized administrator. This email is to prompt actions for overwriting the oldest audit records and implementing recovery measures for the audit storage.

6.1.2 Cryptographic support (FCS)

Cryptographic key and Random bit generation

FCS_CKM.1
FCS_RBG.1

The TOE generates secure cryptographic keys with a strength of at least 112 bits using the key generation algorithm outlined in the following table.

Item	Cryptographic Key	Detail	Cryptographic Key Size	List of Cryptographic

	Generation Algorithm		(bits)	Operations
KEK	PBKDF2	HMAC-SHA256 Salt: 128bits Iteration: 1000	256	- TOE Master Key Decryption - TOE RSA Private Key Decryption
Master Key	Hash_DRBG	Hash: SHA-256	512 (256,256)	- Encryption and decryption for Important information of TOE stored data - Integrity verification of authentication token
Secret Key	Hash_DRBG	Hash: SHA-256	128	- Mutual Authentication between TOE Components
Digital signature key pair	Hash_DRBG	Hash: SHA-256	2048	- Mutual Authentication between TOE Components
Shared Key	ECDH	Curve: P-256 Hash: SHA-256	512 (256,256)	- Data Encryption and Decryption between TOE Components - Data Integrity Verification between TOE Components
Elliptic curve key pair	Hash_DRBG	Hash: SHA-256	256	- Key exchange between TOE components

[Table 6-1] Cryptographic Key Generation Algorithm

Cryptographic key generation using Password-Based Key Derivation Functions (PBKDF2) is utilized only to generate a Key Encryption Key (KEK). Message authentication codes use the HMAC-SHA256 algorithm during this process, with a random 128-bit salt value and a 1000 iteration count applied.

During the initial installation of TOE, a random value generated by the validated cryptographic module's random bit generator is used as the master key. The first 256 bits of the master key are used as a symmetric key for block encryption, while the last 256 bits are used as a MAC key for integrity verification.

The secret key generates a 256-bit key using the validated cryptographic module's random bit generator. The generated secret key is used as the MAC key for the HMAC-SHA256 integrity verification algorithm, used for mutual authentication between the TOE components, such as the SSO server and the SSO agent.

The digital signature key uses the validated cryptographic module's random bit generator to generate a 2048-bit RSA key pair. This key pair is used for the signing and verification key of the RSA-PSS digital signature algorithm for mutual authentication between TOE components like the SSO server and the SSO agent.

The shared key is used for encrypting and decrypting transmitted data between the TOE components, Pass-Ni SSO Server and Pass-Ni SSO Agent. The shared key uses the ‘ECDH elliptic-curve cryptography (Curve: P-256)’ to share mutual cryptographic keys. Of the shared 512-bit cryptographic key, the first 256 bits are used as a symmetric key for block cipher, and the last 256 bits are used as a MAC key for integrity verification.

When generating a cryptographic key using a random bit, a 'hash-function-based deterministic random bit generator (Hash_DRBG)' specified in 'KS X ISO/IEC 18031' is used, and the hash function generates a random bit using SHA-256.

The cryptographic module used by the TOE uses the validated cryptographic module specified in **[Table 1-3]** in which the security and implementation conformance are validated by the Korea Cryptographic Module Validation Program (KCMVP).

Cryptographic key distribution

FCS_CKM.2 The TOE components called Pass-Ni SSO Server and Pass-Ni SSO Agent
FCS_COP.1(4) distribute mutual shared keys, using the Elliptic-curve based key establishment mechanism (ECDH, Curve: P-256)’ described in 'ISO/IEC

11770-3.'

Cryptographic key destruction

FCS_CKM.4 If the TOE is terminated, or when the cryptographic keys loaded onto memory are no longer used, all cryptographic keys loaded onto memory shall be destroyed.

The method of destroying cryptographic keys is to overwrite three times by zero (0).

Cryptographic operation

FCS_COP.1(1) The TOE performs encryption/decryption with symmetric cryptographic (block cipher) algorithm ARIA-256 when conducting encrypted communication between its components or storing TSF data, and it encrypts/decrypts the private key file of server and agent with SEED algorithm. The operation mode uses the CBC, and the generation of IV (Initialization Vectors) is performed using a method shown in the appendix of "NIST SP 800-38A,' that is, it is generated by using the random bit generator provided by the validated cryptographic module. The list of cryptographic operations performed by the block cipher algorithm is shown in **[Table 5-5]**.

FCS_COP.1(2) For the integrity verification of authentication token generated by the TOE, the message authentication code algorithm HMAC-SHA256 is used. The list of cryptographic operations performed using the HMAC-SHA256 message authentication code is as shown in **[Table 5-6]**.

FCS_COP.1(3) The TOE uses the hash function SHA-256 to verify the authentication data of user/administrator and the integrity verification of the TOE. The list of cryptographic operations performed using the SHA-256 hash function is shown in **[Table 5-7]**.

FCS_COP.1(5) During mutual authentication, the TOE employs the digital signature algorithm RSA-PSS (SHA256) for the agent to verify the server. The list of cryptographic operations performed with the RSA-PSS digital signature is shown in **[Table 5-9]**.

The cryptographic module used by the TOE for cryptographic operations uses the validated cryptographic module specified in **[Table 1-3]**, whose security and implementation conformance have been validated through the Korea Cryptographic Module Validation Program (KCMVP).

6.1.3 Identification and authentication (FIA)

Identification and authentication

FIA_UID.2
FIA_UAU.2
The TOE identifies all users (authorized administrators and authorized users) accessing it. All users attempting access cannot use any of the functions of the TOE until the user is identified.

The TOE provides identification and authentication using ID and password. When a user requests the login screen for initial authentication, the TOE generates a nonce value to encrypt the authentication data and prevent a replay attack, then displays the identification and authentication procedure request screen (login screen). Following this, the TOE receives and verifies the ID and password entered by the user through the identification and authentication procedure request screen (login screen).

The TOE issues an authentication token that identifies and authenticates the end-user based on ID / PW. Thereafter, an authentication token-based authentication is provided to end-users to use the service without additional login action when accessing another business system.

FIA_UAU.7 The TOE masks ("● ● ● ● ●") the input password characters and displays them on the screen when inputting confidential information during the authentication procedure. The TOE does not provide feedback (e.g., an invalid user ID or an incorrect password is entered) on the reason for authentication failure; instead, it generates audit data for the authentication failure.

FIA_UAU.4 The TOE ensures the uniqueness of the session by using the disposable nonce value when authentication and identification are performed by the password-based authentication method, thereby preventing the reuse of the authentication data. The TOE ensures the uniqueness of the authentication token, including the nonce value per authentication token. TOE prevents the reuse of the authentication token when the authentication token exceeds the expiration time, including the expiration time information, and cannot be used.

FMT_PWD.1 The TOE enforces the function to change the password when an authorized administrator accesses the security management screen for the first time, or an end-user accesses through the business system for the first time.

The TOE enforces the function to change the password when an administrator and an end-user perform identification and authentication through the initialized password.

Authentication failure handling

FIA_AFL.1(1)
FIA_AFL.1(2)
FAU_GEN.1

The TOE protects itself from malicious user authentication attempts by providing the user account lock function in [identification and authentication]. The TOE does not provide feedback (e.g., an invalid user ID or an incorrect password is entered) on the reason for authentication failure.

If the number of authentication failures reaches the count specified by an authorized administrator (from 3 to 10, the default value being 5), the user identification and authentication function for that account will be disabled. In this case, the disabling of the user identification and authentication function is categorized either as a temporary block (from 5 to 60 minutes) set by the authorized administrator or a block that remains until lifted by an administrator. However, for the top administrator account provided by default at the time of TOE installation, the identification and authentication function will be temporarily disabled(temporarily blocked) for 5 minutes regardless of the response actions configured by the administrator upon reaching the threshold of failed authentication attempts. The TOE processes any authentication attempts through blocked accounts as failures and generates audit data for these events.

An authorized administrator can view or unlock locked user accounts through the administrator or user management function, depending on account type. Once the account lock is lifted, users who successfully complete the [User Identification and Authentication] can normally use the functions provided by the TOE.

Verification of secrets

FIA_SOS.1 The TOE performs an automated inspection that checks whether it meets permission criteria for secrets set by an authorized administrator when an administrator and an end-user create or change a password or when an authorized administrator changes the default password provided at the initial connection.

FMT_PWD.1 The TOE provides authorized administrators with the ability to set ID policies (length, combination rules) and password policies (length, combination rules, restrictions on repeating the same character, restrictions on consecutive character repetition) of administrator and end-user.

Generation and Destruction of authentication token

FIA_SOS.2 Among the TOE components, Pass-Ni SSO Server verifies (identifies and authenticates) the authentication information entered from the user's login screen and generates and issues an authentication token. The authentication token issued is encrypted and passed to the Pass-Ni SSO Agent.

The structure of the authentication token conforms to the 'RFC7519 JSON Web Token (JWT)' standard. The authentication token has three elements: a header, a payload, and a signature, which are separated by a dot (.), and each has a form of 'xxxxxx.yyyyyy.zzzzzz' encoded by base64Url. The detailed structure of the authentication token is specified in [Table 5-10] . The header and payload parts of the authentication token are guaranteed to be integrity, and the payload part is encrypted with the validated cryptographic module to ensure confidentiality.

FIA_SOS.3 The TOE terminates the session and deletes the authentication token loaded on the memory when the TOE receives a logout request from the user or the user is inactive for a certain time interval (inactivity period set by the authorized administrator. Default: 10 minutes) after the login.

When destroying the authentication token, use the method of overwriting three times by zero (0).

TOE Internal mutual authentication

FIA_IMA.1
FCS_CKM.2
FAU_GEN.1

The TOE sets a mutual cryptographic key by the Elliptic Curve Diffie-Hellman (ECDH) (Curve: P-256) to share the mutual cryptographic key between the components called Pass-Ni SSO Server and Pass-Ni SSO Agent. The mutual authentication between the components is performed through the self-implemented authentication protocol using the HMAC-SHA256 integrity verification algorithm and the RSA-PSS digital signature algorithm.

In the self-implemented authentication protocol mechanism, the SSO Server uses the HMAC-SHA256 algorithm to verify the SSO Agent. During the preparative procedure, a 256-bit identification key generated by the validated cryptographic module's random bit generator is loaded in the SSO Server and SSO agent during installation, respectively.

In the self-implemented authentication protocol mechanism, the SSO Agent uses the RSA-PSS (SHA256) algorithm to verify the SSO Server. During the preparative procedure, a 2048-bit RSA key pair is loaded in the SSO Server (private key, public key) and the SSO Agent (public key) during installation, respectively.

The mechanism of the self-implemented authentication protocol for mutual authentication between TOE components is as follows:

1) The SSO agent generates a random bit and creates a digital signature (HMAC-SHA256) using the generated random bit as the secret key, then sends the agent information, random bit, and digital signature value to the SSO server.
2) The SSO server verifies the received data with the digital signature (HMAC-SHA256) using the SSO agent's secret key.
3) The SSO server generates a random bit and creates a digital signature

(RSA-PSS) with the random bit of the SSO agent and the SSO server as a private key, then sends the server information, random number, and digital signature value to the SSO agent.

4) The SSO agent verifies the received data with the digital signature (RSA-PSS) using the SSO server's public key.

The TOE generates audit data for success or failure events in the mutual authentication between components.

6.1.4 Security management (FMT)

Once the TOE is properly installed, an authorized administrator can access the TOE Security Management Interface (GUI) through a web browser (e.g., Google Chrome) from a management PC with an explicitly allowed IP address. The TOE permits access to the Security Management Interface (HTTPS) only if the user attempting to connect completes the identification and authentication process enforced by the TOE successfully.

Security roles

FMT_SMR.1 The roles of an authorized administrator of the TOE are divided into two types: top administrator and monitoring administrator.

a) Top Administrator: an authorized administrator who has full authority and has been granted the authority of all the security management functions provided by the TOE.

b) Monitoring Administrator: an administrator who has been granted the authority to perform the inquiry functions among the security management functions provided by the TOE.

The ID of the Monitoring Manager is not provided by default in the TOE, but only the role is defined. Therefore, the Top Administrator must register a new administrator when needed and grant the necessary permissions for operation.

FAU_GEN.1 The TOE generates audit data for management actions when the authorized administrator performs the security management function.

6.1.4.1 Link System management

The TOE performs the [Link System Management] function for managing the link system (business system) where the SSO agent is installed.

Link System Management

FMT_MOF.1
FMT_SMF.1
FMT_MTD.1

The TOE provides the authorized administrator with the [Link System Management] function to register, modify, or delete the business system. For an end-user to login to the business system using the Single Sign-On service, the business system must be registered by the authorized administrator using this function.

The authorized administrator manages the business system's ID, system name, character set, business system identification code, usage status, domain, and IP information through the security management interface. The TOE uses the business system information registered by the authorized administrator to provide integrated Single Sign-On service to end-users.

6.1.4.2 Policy management

The TOE performs the [policy management of user ID and password], [policy management of user and administrator], and [policy management of audit violation] functions for managing the operational policy.

Policy management of user ID and password

FMT_MOF.1
FMT_SMF.1
FMT_MTD.1
FMT_PWD.1

The TOE provides the authorized administrator with the [policy management of user ID and password] function of managing the allowable characters, combination rule, and length for each ID and password.

The TOE provides the authorized administrator with the function to manage the following ID policies. When creating the administrator and user ID, TOE enforces each verification criteria of the following ID policies:

a) ID character count: Minimum count (default: 6, input range: 5 to maximum count), Maximum count (default: 20 characters, input range: minimum count to 20 characters)

b) ID character inclusion conditions
- Inclusion of uppercase letters: Enabled/Disabled (default: disabled)
- Inclusion of lowercase letters: Enabled/Disabled (default: enabled)
- Inclusion of numbers: Enabled/Disabled (default: disabled)
- Inclusion of special characters: Enabled/Disabled (default: disabled)

The TOE provides the authorized administrator with the function to manage the following password policies. When creating or changing the administrator and user password, TOE enforces each verification criteria of the following password policies:

a) Password character count: Minimum count (default: 9 characters, input range: 6 characters to maximum count), Maximum count (default: 20 characters, input range: minimum count to 99 characters)

b) Password character inclusion conditions
- Inclusion of uppercase letters: Enabled/Disabled (default: Disabled)
- Inclusion of lowercase letters: Enabled/Disabled (default: Enabled)
- Inclusion of numbers: Enabled/Disabled (default: Enabled)
- Inclusion of special characters: Enabled/Disabled (default: Enabled)

c) Identical character repetition in password: Enabled/Disabled, Repeat count (default: 3 times, input range: 2-9 times)

d) Consecutive character sequence in password: Enabled/Disabled, Consecutive count (default: 3 times, input range: 2-9 times)

e) Consecutive characters in keyboard layout in password: Enabled/Disabled, Consecutive count (default: 4 times, input range: 2-9 times)

f) Limit on recent password change: Enabled/Disabled, Limit count (default: 1 time, input range: 1-10 times)

Policy management of end-user and administrator

FMT_MOF.1
FMT_SMF.1
FMT_MTD.1

The TOE provides the authorized administrator with the [policy management of end-user and administrator] function of managing the number of failed password entries (number of authentication failures) and

FIA_AFL.1 FTA_TSE.1	the password change interval for each end-user and administrator. The TOE provides the authorized administrator with the function to manage the following policies for end-users. TOE enforces each verification criteria of these end-user policies during user identification and authentication: a) Block on login failure: Enabled/Disabled (default: Enabled), Number of failures (default: 5 attempts, input range: 3-10 attempts), Block/Temporary block (default: Temporary block), Block duration (default: 5 minutes, input range: 5-60 minutes) b) Password change interval: Enabled/Disabled (default: Enabled), Change interval (default: 60 days, input range: 1-365 days) c) Automatic logout: Enabled/Disabled (default: Enabled), Timeout duration (default: 10 minutes, input range: 1-60 minutes) d) Limit on concurrent logins: Enabled/Disabled (default: Disabled) e) Authentication token validity period: Validity period (default: 300 minutes, input range: 30-1000 minutes) The TOE also provides the authorized administrator with the function to manage the following policies for administrators. TOE enforces each verification criteria of these administrator policies during administrator identification and authentication. a) Block on login failure: Number of failures (default: 5 attempts, input range: 3-10 attempts), Block/Temporary block (default: Temporary block), Block duration (default: 5 minutes, input range: 5-60 minutes) b) Password change interval: Change interval (default: 60 days, input range: 1-365 days) c) Number of access IPs: Number of IPs (default: 2, input range: 1-99)
FMT_MOF.1 FMT_SMF.1 FMT_MTD.1 FAU_SAA.1	**Policy management of audit violation** The TOE examines audit-targeted events to detect potential security breaches and notifies the authorized administrator via email. In this regard, TOE provides a [policy management of audit violation] function, allowing the management of email notification settings for each security breach event.

In the TOE management functions, the events for which email notification settings for security breach events can be managed are as follows:

- Exceeding administrator authentication failure count (default: Enabled)
- Exceeding user authentication failure count (default: Enabled)
- Server self-test failure (default: Enabled)
- Agent self-test failure (default: Enabled)
- Exceeding audit repository threshold (default: Enabled)
- Audit repository saturation (default: Enabled)

6.1.4.3 End-user management

The TOE performs [End-user management] and [Administrator management] functions for the management of TOE users.

End-user management

FMT_MOF.1 The TOE provides the [End-user management] function for managing end-
FMT_SMF.1 users who use the Single Sign-On service. The authorized administrator can
FMT_MTD.1 inquire about currently connected users through the security management
interface and the forced logout of selected users.

The TOE provides authorized administrators with the function to inquire about the registered end-user and register, modify, or delete. The authorized administrator can manage the end-user's name, phone number, email, and enable/disable (active/inactive status of the end-user). Additionally, TOE provides the functionality to reset the passwords for the end-users who have lost their passwords and to unlock the authentication and identification functions disabled due to exceeding the preset number of authentication failures (default: 5 attempts).

The TOE also provides the capability to change their own passwords to the end-user.

Administrator management

FMT_MOF.1 The TOE provides the [administrator management] function to
FMT_SMF.1 inquire/register/modify/delete administrators that can access through the

FMT_MTD.1 security management interface. The TOE provides the top administrator as a default, and the authorized administrator can inquire about the registered administrator, register the new administrator, or delete the existing administrator.

The authorized administrator can manage the name, phone number, e-mail, start/end date of account activation, notification email reception, and enable/disable (active/inactive status of the end-user). If the start/end date of use is specified, the security management interface can be accessed only for a specified period. When disabled, security management access (identification and authentication) of the administrator is restricted.

FMT_SMR.1 The authorized administrator can select the administrator role defined in the TOE at administrator registration/modification. There are two types of authorized administrators of the TOE: top administrator and monitoring administrator.

FTA_TSE.1 The TOE provides the function to register/modify/delete access permitted IPs for each administrator as the number of access permitted IPs (default: 2) set in the [policy management of administrator] function through the [administrator management] function. The administrator can only access the registered terminal as the access permitted IP when requesting identification and authentication.

6.1.4.4 System management

The TOE performs [System check], [BATCH management], and [Code management] functions for managing the TOE system.

System check

FMT_MOF.1 The TOE performs self-tests during initial start-up and periodically to
FMT_SMF.1 demonstrate the correct operation of the TSF. In addition, provides the
FPT_TST.1 authorized administrator with a [system check] function to perform self-
FAU_GEN.1 tests and tests of external entities, if necessary.

The self-test consists of server self-test and agent self-test. The server self-test and agent self-test are conducted upon the request of an authorized administrator, performing integrity verification and self-test of the cryptographic module for each SSO server and agent. Additionally, audit data is generated based on the results of these tests.

BATCH management

FMT_SMF.1 The TOE provides the [BATCH management] function to manage
FMT_MTD.1 functions executed periodically during TOE operation. The authorized
administrator can set the execution cycle and enable the status of the
following functions through the security management interface.

a) Garbage Collection: Clear memory by the garbage collection
b) Database Capacity Check: Audit trail storage capacity inspection
c) Server Self-Test: Integrity verification, cryptographic module self-test
d) Agent Self-Test: Integrity verification, cryptographic module self-test
e) Statistics Generation: Creation of user access statistical data

Code management

FMT_SMF.1 The TOE provides the [code management] function for managing codes
FMT_MTD.1 used during TOE operations. Authorized administrators can manage audit
types, error messages (result codes), email types, character set types,
business system audit types, and system audit types through the security
management interface.

6.1.4.5 Audit view and statistics

The TOE performs functions such as [system audit inquiry], [link system audit inquiry], [administrator access and activity audit inquiry], [end-user access and activity audit inquiry], and [statistics inquiry].

System audit inquiry

FAU_SAR.1 The TOE generates audit data for events related to the operation of its
FMT_MOF.1 security functions. The TOE provides authorized administrators with the
FMT_SMF.1 [System audit inquiry] feature to check the operational status of the TOE.

The types of system audit items that authorized administrators can inquire through the security management interface are as follows:

- Server startup and shutdown
- Server module testing, integrity verification, cryptographic module self-test
- Issuance, storage, verification, and disposal of authentication tokens
- Mail sending
- DB volume check
- Data cleanup
- Generation of statistical information

FAU_SAR.3

Following **[Table 5-3]**, the TOE organizes the audit data in descending order based on the time of occurrence. Authorized administrators can sort the retrieved audit list in descending or ascending order based on serial number, status, IP, type, and time of occurrence.

Link system audit inquiry

FAU_GEN.1
FAU_SAR.1
FMT_MOF.1
FMT_SMF.1

The TOE generates audit data on the results of interoperating with the SSO agents deployed and operating in the link system. The TOE provides authorized administrators with the [Link system audit inquiry] function to inquire about these link system interoperating records. The types of link system audit items that authorized administrators can inquire through the security management interface are as follows:

- Agent startup and shutdown
- Agent module testing, integrity verification, cryptographic module self-test
- Login request, logout request
- Key exchange request
- Authentication token issuance request, user information request
- Login check request
- link system list request
- Security policy information request
- User authentication request

FAU_SAR.3

Following **[Table 5-3]**, the TOE organizes the audit data in descending order based on the time of occurrence. Authorized administrators can sort the retrieved audit list in descending or ascending order based on serial number, type, name of the link system, and time of occurrence.

Administrator access and activity audit inquiry

FAU_GEN.1
FAU_SAR.1
FMT_MOF.1
FMT_SMF.1

The TOE generates audit data for activity results when an authorized administrator accesses the security management interface or performs security management functions. The TOE provides a [Administrator access and activity audit inquiry] function to inquire about the access and activity history of authorized administrators.

FAU_SAR.3

Following **[Table 5-3]**, the TOE organizes the audit data in descending order based on the time of occurrence. Authorized administrators can sort the retrieved audit list in descending or ascending order based on serial number, status, ID, name, and time of occurrence.

End-user access and activity audit inquiry

FAU_GEN.1
FAU_SAR.1
FAU_SAR.3
FMT_MOF.1
FMT_SMF.1

The TOE generates audit data for activity results of end-user actions such as access (identification and authentication), password change, and logout. The TOE provides a [End-user access and activity audit inquiry] function to inquire about the access and activity history of end-users.

FAU_SAR.3

Following **[Table 5-3]**, the TOE organizes the audit data in descending order based on the time of occurrence. Authorized administrators can sort the retrieved audit list in descending or ascending order based on serial number, status, ID, name, and time of occurrence.

Statistics inquiry

FMT_SMF.1

The TOE provides annual and monthly user access statistics based on the user access audit data generated during the operation of the TOE.

6.1.5 Protection of the TSF (FPT)

TSF data transfer protection

FPT_ITT.1
FCS_CKM.2
FCS_COP.1(1)

The TOE sets a mutual cryptographic key by the Elliptic Curve Diffie-Hellman (ECDH) (Curve: P-256) to share the mutual cryptographic key between the components called Pass-Ni SSO Server and Pass-Ni SSO Agent. The mutual authentication between the components is performed through the self-implemented authentication protocol using the RSA-PSS (SHA256) algorithm and the HMAC-SHA256 algorithm. The sharing of the cryptographic key between the TOE components is performed when the Pass-Ni Agent is started up.

The transmission data between the TOE components is sent and received by encryption with the validated algorithm (default: ARIA256) using the shared cryptographic key in the above key-sharing process.

Protection of stored TSF data

FPT_PST.1

The TOE protects the information specified in '**[Table 5-13]** ' using access control and encryption (hashing, symmetric key) to safeguard against unauthorized exposure and alteration. TSF data stored in the DBMS is protected by the DBMS's identification, authentication, and access control features to prevent unauthorized access. TSF data is encrypted using the validated cryptographic module's validated cryptographic algorithm. TSF data of configuration files stored in the filesystem is encrypted and protected using the validated cryptographic module's validated cryptographic algorithm.

The passwords of end-users and administrators are encrypted using the validated cryptographic module's validated hash function (default: SHA256), including a random 32-character string salt. The TOE uses a message authentication code (HMAC-SHA256) to ensure the integrity of the authentication token. The authentication token stored in the server storage are encrypted using the validated cryptographic algorithm (ARIA256). The secret key is used for mutual authentication between the server and agent, and the shared cryptographic key during the agent's startup is encrypted using the validated cryptographic algorithm (ARIA-256) when stored in the server storage. Additionally, the DBMS ID/password and SMTP ID/password stored in the filesystem are encrypted and stored using the validated cryptographic algorithm (ARIA-256).

The Key Encryption Key (KEK) is a cryptographic key generated from the entered password by the administrator upon the start-up of the TOE, using the PBKDF2 key derivation function. The KEK is used to decrypt the DEK (master key), which is employed for the protection of TSF data. The PBKDF2 key derivation function uses the HMAC-SHA256 algorithm, with a salt value of random 128 bits, and 1000 iterations are applied.

The cryptographic key loaded into memory is protected against memory dump attacks by being XOR-operated with a random Salt value upon loading. The Salt value is the generated random bit using the validated random bit generator at server startup, and to protect against string dump attacks, it is loaded into memory in binary data (byte array) format. The cryptographic key is restored to its plaintext form by again XOR-operating it with the Salt value when it is used for encryption/decryption operations. Once used, the restored cryptographic key is destroyed following the method specified in FCS_CKM.4 for cryptographic key destruction.

Self-test

FPT_TST.1
FAU_ARP.1
FAU_GEN.1
FAU_SAA.1

The TOE conducts server self-tests after the initial TOE startup and periodically (default interval: 24 hours) during regular operation to demonstrate the correct operation of its security functions. An authorized administrator can set the test interval through the TOE security management interface. The self-test performs integrity check and cryptographic module self-tests for the SSO server and agent.

Integrity check verifies the integrity of the TOE's executable code and configuration files. The TOE generates hash values for the items subject to integrity checks at each designated inspection interval. It compares them with the stored hash values (reference values) during initial installation or configuration changes. If a breach of integrity is detected, the TOE notifies the authorized administrator via the set email and generates audit data for the incident. The hash algorithm used for integrity checks is the SHA-256 hash function provided by the validated cryptographic module. The TOE executable code and configuration files subject to integrity checks are as follows.

Component	Type	Target Files
SSO Server	Executable Code	passni-sso-common-5.0.{distribution version}.jar passni-sso-connect-cache-5.0.{distribution version}.jar passni-sso-connect-dbms-5.0.{distribution version}.jar passni-sso-filter-5.0.{distribution version}.jar passni-sso-sso-adm-5.0.{distribution version}.jar passni-sso-sso-com-5.0.{distribution version}.jar passni-sso-sso-token-5.0.{distribution version}.jar passni-sso-sso-user-5.0.{distribution version}.jar
	Configuration File	common-config.xml dbms-config.xml passni-config.xml
	Cryptographic Library	{crypto-installdir}/*.so
SSO Agent	Executable Code	passni-sso-agent-5.0.{distribution version}.jar
	Configuration File	pni5-config.xml
	Cryptographic Library	{crypto-installdir}/*.dll {crypto-installdir}/*.so

[Table 6-2] Integrity check target files

The cryptographic module self-test verifies the proper functioning of the validated cryptographic module, whose security and conformity implementation have been confirmed through the Korea Cryptographic Module Validation Program (KCMVP). This test is conducted by calling the self-test API interface of the validated cryptographic module. If an error is detected, the TOE notifies the authorized administrator via the set email and generates audit data for the incident.

The authorized administrator can perform the server self-test through the TOE security management interface. If the test fails, the TOE notifies the authorized administrator via the designated email and generates audit data for the incident.

The authorized administrator can take response actions such as ignoring the issue or rebooting the server. Audit data is generated for all these events.

6.1.6 TOE access (FTA)

Administrator session management

Once the TOE is properly deployed/installed, it provides a web-based interface accessible by the authorized administrator using a web browser installed on their management PC. The TOE only allows access to the management access interface (HTTPS) if the user attempting to connect successfully completes the administrator identification and authentication process.

FTA_TSE.1 The TOE allows or blocks access to the management access interface based on the authorized administrator's access IP, status (enable/disable of the administrator), and usage period (start date/end date). The TOE denies the management access session if the administrator attempting to access the management access interface is not using a registered access IP if the usage period has not yet begun or has expired, or if the status is disabled.

FTA_SSL.5 After the authorized administrator successfully login to the TOE's management access interface (Web UI), the TOE terminates the session interacting with the authorized administrator if the allowed inactivity period is exceeded. This inactivity period is set by default to 10 minutes and cannot be changed. When the session is terminated, if the authorized administrator attempts to use the security management interface again, they are redirected to the management access initial screen (login screen), and the administrator must perform administrator re-authentication (user identification and authentication).

FTA_MCS.2 The TOE limits the maximum number of concurrent sessions for administrator management access to one(1). Consequently, the same user cannot have more than one simultaneous connection to the management access interface. The TOE terminates the previous session when the same account or administrator with the same authority concurrently accesses the management interface. However, for the monitoring administrator, concurrent logins with other administrative accounts are permitted.

FAU_GEN.1	The TOE generates audit data for the outcomes of these events, namely the results of [Administrator session management].
	User session management
FTA_SSL.5 FAU_GEN.1	The TOE provides the authorized administrator the function to set the inactivity time for the end-user (from 1 to 60 minutes, or 'Off', with a default of 10 minutes). If an end-user is inactive for the duration of the inactivity time set by the authorized administrator following login, the TOE will terminate that session and generate audit data for the action. Upon session termination, the TOE blocks an end-user from accessing personalized feature screens, such as the password change screen, and the end-user must perform re-authentication (end-user identification and authentication).