RATHON TECH

# Security Target Specification (ST)

**Rathon-SSO v4.0**

**Rathon Tech Co., Ltd.**

# Revision History

| Version | Date | Content |
|---|---|---|
| 1.0 | February 27, 2025 | Initial creation |
| 1.1 | February 13, 2026 | Cryptographic Key Functions, Vulnerabilities, Reference Standards, and Typographical Corrections |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

# Table of Contents

# List of Figures

# List of Tables

# 1. ST Introduction

## 1.1. Security Target Reference

**[Table 1] Security Target Reference**

| | |
|---|---|
| Title | Rathon-SSO v4.0 ST_EN |
| Version | v1.1 |
| Author | RathonTech Co., Ltd. Research Institute |
| Publication Date | February 27, 2025 |
| Common Criteria | Common Criteria for Information Technology Security Evaluation CC:2022 Release 1<br>- Part 1: Introduction and General Model, CC:2022 r1 (CCMB-2022-11-001, November 2022)<br>- Part 2: Security Functional Components, CC:2022 r1 (CCMB-2022-11-002, November 2022)<br>- Part 3: Security Assurance Components, CC:2022 r1 (CCMB-2022-11-003, November 2022)<br>- Part 4: Framework for Evaluation Methodology and Activities, CC:2022 r1 (CCMB-2022-11-004, November 2022)<br>- Part 5: Pre-defined Packages of Security Requirements, CC:2022 r1 (CCMB-2022-11-005, November 2022)<br>- Common Criteria Evaluation Methodology for Information Security Systems, CCMB-2022-11-006, CEM:2022 Revision 1, November 2022.<br>- Common Criteria for Information Security Systems CC:2022 R1 Errata and Interpretation for CC:2022 (Release 1) and CEM:2022 (Release 1), Version 1.2, CCMB-2025-001, October 2025. |
| Evaluation Assurance Level | EAL1+(ATE_FUN.1) |
| Protection Profile | National Integrated Authentication Protection Profile V3.1 |
| Product Category | Single Sign On (SSO) |

## 1.2. TOE reference

**[Table 2] TOE reference**

| Category | | Description |
|---|---|---|
| TOE Name | | Rathon-SSO v4.0 |
| TOE Detailed Version | | v4.0.1 |
| Components | SSO Server | Rathon-SSO Server v4.0.1 |
| | SSO Agent | Rathon-SSO Agent v4.0.1 |
| Manuals | Preparative Procedures | Rathon-SSO v4.0 Preparation Procedure v1.1 |
| | Operational Guidance | Rathon-SSO v4.0 Operating Manual v1.1 |
| Developer | | RathonTech Co., Ltd. Research Institute |

# 1.3. TOE overview

## 1.3.1. Single Sign On overview

Rathon-SSO v4.0 (hereinafter referred to as the 'TOE') is used for the purpose of providing services to users through a single login (Single Sign-On) without additional login actions when accessing various business systems. The TOE performs user identification and authentication, authentication token issuance, and validity verification functions in accordance with the user authentication policy.

The TOE provides a user login function using an ID/password-based authentication method, issues an authentication token during user login, and verifies the issued authentication token when the user accesses another business system after login. ID and password-based authentication functions for authorized administrators and authorized general users are mandatorily required in the TOE; however, for general users, these functions are applied only when the TOE, rather than an external authentication system, provides the authentication function in the initial authentication phase of single sign-on.

The primary security functions provided by the TOE include user identification and authentication functions and the issuance, storage, verification, and destruction of authentication tokens. The TOE uses a validated cryptographic module whose security and implementation conformance have been verified through the Korea Cryptographic Module Validation Program (KCMVP) when generating authentication tokens and performing authentication token-based user single sign-on.

## 1.3.2. TOE type and scope

The TOE is a 'Single Sign-On' (SSO) system that enables access to various business systems through a single user login, and the TOE components are provided in the form of software.

The TOE components consist of the Rathon-SSO Agent (hereinafter referred to as the 'SSO agent') and the Rathon-SSO Server (hereinafter referred to as the 'SSO server'). The TOE is composed of a server that performs functions such as user login processing, authentication token management, and policy configuration, and an agent that is installed in each business system to perform functions such as authentication token issuance and authentication token verification requests. In addition, the agent is provided in the form of an 'API type' composed of library files.

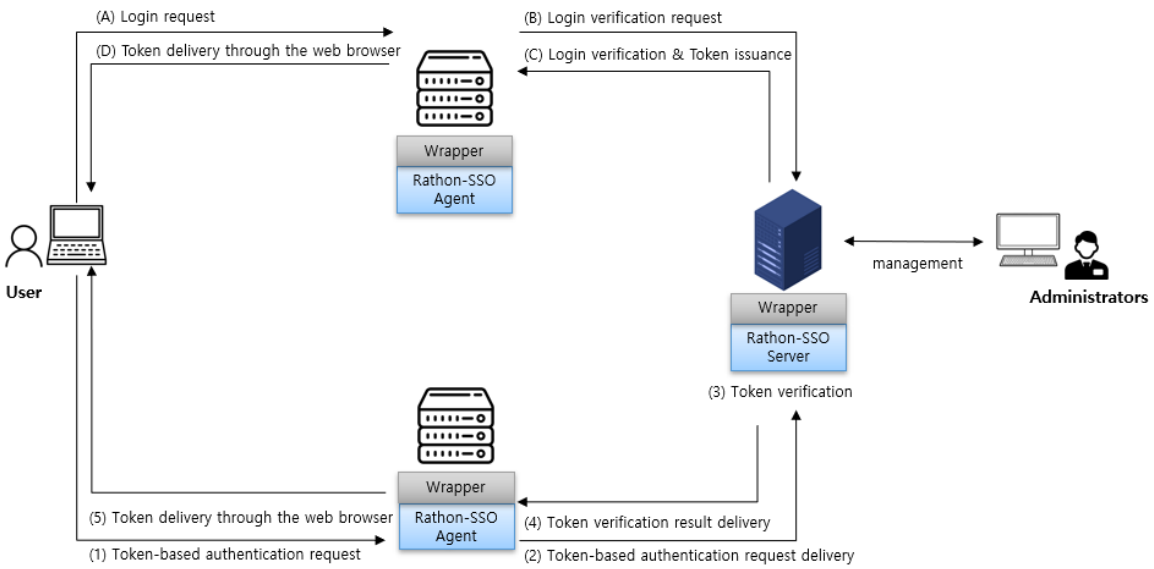
## 1.3.3. TOE usage and major security features

The TOE performs user identification and authentication functions to provide services to users through a single login (Single Sign-On) without additional login actions when accessing various

business systems.
The TOE provides security audit functions that record and manage major events as audit data during operation as security and management functions, TSF protection functions such as protection of data stored in repositories controlled by the TSF, and TSF self-tests. In addition, the TOE provides identification and authentication functions such as authentication failure handling and mutual authentication between TOE components, cryptographic support functions such as cryptographic key management and cryptographic operations for authentication token issuance, security management functions for managing security functions and configuration settings, and TOE access functions for managing access sessions of authorized administrators.
Additionally, authentication tokens require confidentiality and integrity, and the TOE executable code requires integrity. The general user identification and authentication procedure of the TOE is as shown in [ [Figure 1] User identification and authentication procedure ], and the detailed execution procedure is divided into an initial authentication phase based on ID/password and an authentication token-based business system authentication phase.

The user identification and authentication process is divided into an initial authentication phase using ID/password-based authentication and an authentication token-based authentication phase in which the user accesses business systems using the authentication token issued during the initial authentication process.
First, the execution procedure of the initial authentication phase is as follows. The user requests login using an ID/password, and the SSO agent that receives the login request message sends a login verification request to the SSO server to check whether the user is legitimate. Upon receiving the login verification request, the SSO server performs login verification directly using the user information stored in the DBMS. If the login verification result is valid, the SSO server issues an authentication token. The SSO agent delivers the token issued by the SSO server to the user.

The authentication token-based authentication phase is performed only when an authentication token has been normally issued through the initial authentication phase. When the user uses business system services, the issued authentication token is delivered to the SSO agent installed in the corresponding business system, and the SSO agent that receives the token verifies the validity of the authentication token through interaction with the SSO server.




**[Figure 1] User Identification and Authentication Procedure**

The step-by-step operation procedure of the user identification and authentication process is as shown in [ [Table 3] Operation procedure by authentication phase ].

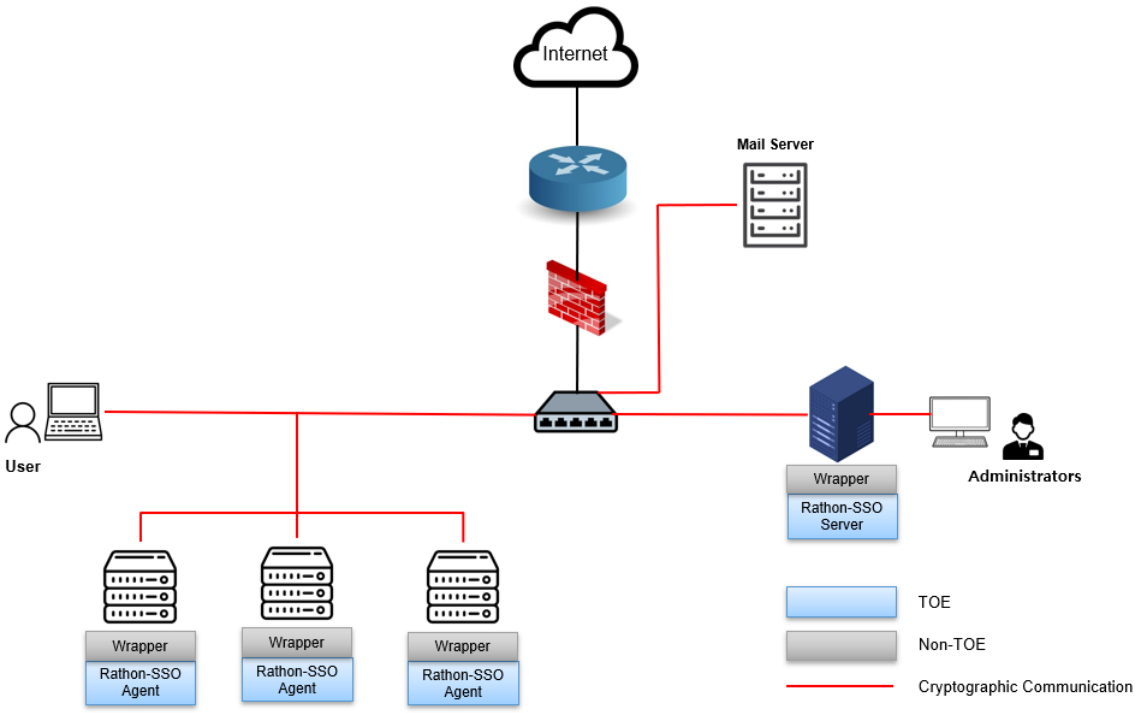
| Authentication phase | Operation Procedure |
|---|---|
| Initial Authentication | (A) Login request → (B) Login verification request → (C) Login verification & token issuance → (D) Token delivery through the web browser |
| Token-Based Authentication | (1) Token-based authentication request → (2) Token-based authentication request delivery → (3) Token verification → (4) Token verification result delivery → (5) Token delivery through the web browser |

**[Table 3] Operation procedure by authentication phase**

In addition, the subjects responsible for authentication token issuance, storage, and verification are as follows.

- Subject responsible for authentication token issuance: Rathon-SSO Server
- Authentication token storage location: user PC web browser, Rathon-SSO Server
- Subject responsible for authentication token verification: Rathon-SSO Server

### 1.3.4. Non-TOE and TOE Operational Environment




**[Figure 2] TOE Operational Environment**

The TOE operational environment is as shown in [ [Figure 2] TOE Operational Environment ]. The TOE operational environment consists of an SSO server and an SSO agent.
The SSO server provides user login verification directly using user information stored in the DBMS, authentication token management, and policy configuration. The SSO agent performs user login verification requests to the SSO server and authentication token issuance and verification request functions, and operates by being installed in each business system. In addition, the SSO agent is provided in an 'API type' composed of library files.

Authorized administrators access the SSO server through a web browser to perform security management. In the TOE operational environment, a wrapper may be used to ensure compatibility with business systems, and the wrapper is excluded from the TOE scope.
Encrypted communication is performed in the communication sections between TOE components, and encrypted communication using TLS v1.3 or higher is also performed when communication between the mail server and TOE components is required.
As an external entity required for operating the TOE, a mail server is used to notify authorized administrators in cases such as administrator authentication failure or predicted audit data loss.

The minimum system requirements for installing and operating the TOE are as follows.

**[Table 4] Non-TOE Hardware Requirements**

| TOE | Classification | Item | Operational Environment Requirements |
|---|---|---|---|

| | | | |
|---|---|---|---|
| SSO Server | H/W | CPU | AMD Ryzen 5 5600G with Radeon Graphics 4.464 GHz or higher |
| | | Memory | 16 GB or more |
| | | HDD | Required storage space for TOE installation: 500 GB or more |
| | | NIC | Ethernet 100/1000 Mbps * 1 Port or more |
| SSO Agent | H/W | CPU | AMD Ryzen 5 5600G with Radeon Graphics 4.464 GHz or higher |
| | | Memory | 16 GB or more |
| | | HDD | Required space for TOE installation: 500 GB or more |
| | | NIC | Ethernet 100/1000 Mbps * 1 Port or more |

The operating systems on which the TOE operates are as follows.

**[Table 5] Supported Operating Systems for the TOE**

| TOE | Operating System |
|---|---|
| SSO Server | Ubuntu 24.04.3 (Kernel 6.8.0) 64 bit |
| SSO Agent | Ubuntu 24.04.3 (Kernel 6.8.0) 64 bit |

The non-TOE software that is not included in the TOE scope but is required for normal operation is as follows.

**[Table 6] Non-TOE Software**

| TOE | Item | S/W | Details |
|---|---|---|---|
| SSO Server | WAS | Apache Tomcat 11.0.18 | A Java-based web application server (WAS) required for normal operation of the TOE |
| | DBMS | PostgreSQL 17.7 | A repository for TOE policy configuration and audit data |
| | WAS | Apache Tomcat 11.0.18 | A Java-based web application server (WAS) required for normal operation of the TOE |

The following additional systems are required in the IT environment for TOE operation.

**[Table 7] TOE Supported IT Environment**

| Category | Description and Role |
|---|---|
| Mail Server<br>(SMTP Server) | Send an alert email to authorized administrators when continuous authentication failures by users, audit log repository threshold exceedance and saturation, TOE self-test failure, or TSF data integrity compromise is detected. |

The minimum requirements for the administrator system for security management are as follows.

**[Table 8] Administrator operational environment requirements**

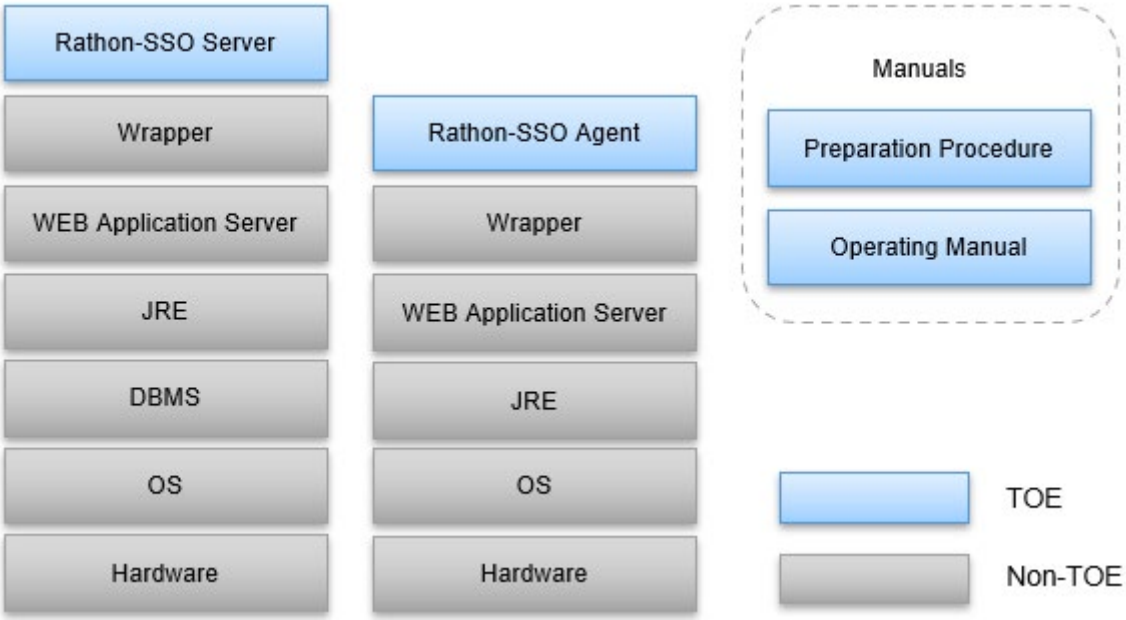
| Category | Classification | Item | Description |
|---|---|---|---|
| Administrator PC | S/W | Browser | Chrome 145.0 (64 bit) |

## 1.4. TOE description

This chapter describes the physical scope and logical scope of the TOE.

### 1.4.1. Physical scope of the TOE

The TOE consists of the Rathon-SSO Server (hereinafter referred to as the 'SSO server'), the Rathon-SSO Agent (hereinafter referred to as the 'SSO agent'), the preparation procedures document, and the user operation manual document.




**[Figure 3] Physical Scope of the TOE**

The Rathon-SSO v4.0 product includes the SSO server and SSO agent developed by RathonTech Co., Ltd., and the Preparation Procedure Manual and User operation Guidance Manual are also provided.

**[Table 9] Product and TOE Physical Configuration**

| Category | | Type | Form | Deployment |
|---|---|---|---|---|
| **TOE Name** | | Rathon-SSO v4.0 | S/W | CD |
| **TOE Detailed Version** | | v4.0.1 | | |
| TOE Component | SSO Server | Rathon-SSO Server v4.0.1<br>: Rathon-SSO_Server-v4.0.1.war | | |
| | SSO Agent | Rathon-SSO Agent v4.0.1<br>: Rathon-SSO_Agent-v4.0.1.war | | |
| Manual | Preparation Procedure | Rathon-SSO v4.0 Preparation Procedure v1.1<br>: Rathon-SSO v4.0 Preparation Procedure _v1.1.pdf | Electronic Document (PDF) | |
| | Operating Manual | Rathon-SSO v4.0 Operating Manual v1.1<br>: Rathon-SSO v4.0 Operating Manual _v1.1.pdf | | |

Information on the validated cryptographic modules used by the TOE is as follows.

**[Table 10] Validated Cryptographic Modules Used by TOE**

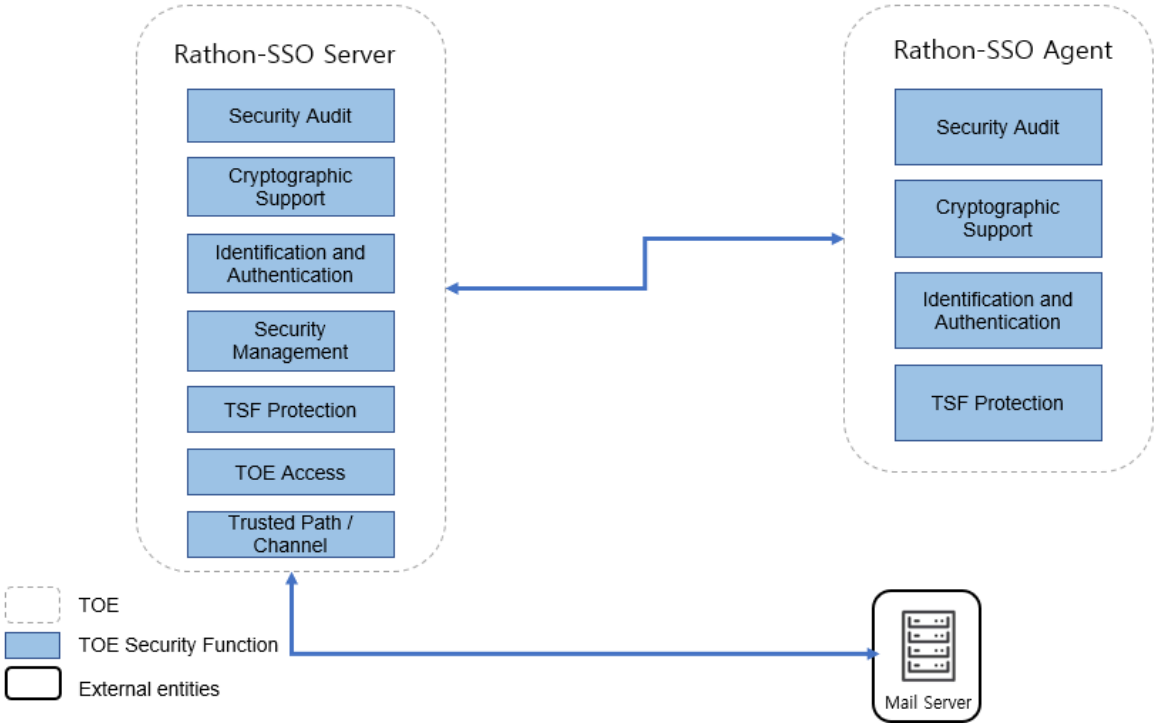
| Cryptographic Module Name | Security Level | Certification Number | Developer | Verification Date | Expiration Date |
|---|---|---|---|---|---|
| RTJCrypto V1.0 | 1 | CM-281-2030.10 | RathonTech Co., Ltd. | October 24, 2025 | October 24, 2030 |

The third-party libraries used for performing the security functions of the TOE are as follows.

**[Table 11] TOE 3rd Party Software**

| TOE | Item | S/W | Purpose |
|---|---|---|---|
| SSO Server | Java | JRE 21.0.10+7 | Used for TLS v1.3 encrypted communication with the SSO agent. |
| SSO Agent | Java | JRE 21.0.10+7 | Used for TLS v1.3 encrypted communication with the SSO server. |

### 1.4.2. Logical scope of the TOE




**[Figure 4] Logical Scope of the TOE**

■ **Security audit (FAU)**

The SSO server generates audit data for accountability of security-related events. The audit data generated by the SSO server records the date and time of the event, event type, identity of the subject, and event result (success or failure). All audit data is stored in the DBMS.

Authorized administrators can view audit data through the administrator interface and search audit data by applying descending order based on the date and time of the event using the event date AND event type as criteria.

When the audit data storage capacity reaches a predefined fixed threshold, a warning email is sent to the administrator. In addition, when the audit data storage becomes full, audit events are not recorded and a warning message is sent to the administrator via email.

Additionally, when the following potential violations are detected, a warning message is sent to the administrator via email.

- List of potential violations [

  When administrator authentication attempts fail consecutively for the defined number of times (fixed value: 5 times),

  When general user authentication attempts fail consecutively for the defined number of times (fixed value: 5 times),

  When a validated cryptographic module self-test fails within the SSO server,

  When integrity verification fails within the SSO server,

  When a process test fails within the SSO server,

  When a validated cryptographic module self-test fails within the SSO agent,

  When integrity verification fails within the SSO agent,

  When a process test fails within the SSO agent,

  When the audit data repository threshold (fixed value: 80%) is exceeded,

  When the audit data repository reaches saturation (fixed value: 90%) ] cumulative or combination

The SSO agent transmits audit data to the SSO server for recording when audit events occur, such as success or failure of user identification and authentication, or integrity verification of the SSO agent.

■ **Cryptographic support (FCS)**

The TOE uses a validated cryptographic module (RTJCrypto V1.0, CM-281-2030.10), whose security and implementation conformance have been verified through the Korea Cryptographic Module Validation Program (KCMVP), to manage cryptographic keys, perform cryptographic operations, and generate random bits for communication between the SSO server and the SSO agent.

The TOE securely generates cryptographic keys based on the validated random number generation mechanism compliant with standards, 'Hash_DRBG(SHA-384)'. The generated

cryptographic keys include an integrity verification key (256 bit), a data encryption key (256 bit), an authentication token encryption key (128 bit), and a session key (128 bit). In addition, public/private key pairs (RSAES and RSA-PSS, 3072 bit) of the SSO server and SSO agent are respectively generated and used for session key protection and digital signatures, and the KEK for DEK protection is securely generated through a PBKDF2 (SHA-256)-based key derivation function.

Cryptographic operations are performed using validated cryptographic algorithms compliant with standards for KEK key derivation for DEK encryption (PBKDF2, SHA-256), TOE component integrity verification (HMAC, SHA-256), one-way encryption of user passwords (SHA-256), data and authentication token protection (ARIA/CBC, 128/256 bit), transmission section security (RSAES, SHA-256), and mutual authentication between the SSO server and the SSO agent (RSA-PSS, SHA-256).
All cryptographic keys used in the SSO server and SSO agent are securely generated, managed, and destroyed, and upon destruction, the key values are overwritten with '0' three times for zeroization.

Furthermore, the TSF performs deterministic random bit generation using the Hash_DRBG (SHA-384) algorithm in accordance with TTAK.KO-12.0331.
When the reseed count limit (1000 times) is reached, the DRBG is reseeded using a software-based entropy source (new SecureRandom().nextBytes()) in accordance with TTAK.KO-12.0235/R2 to update the internal state.
The TSF securely seeds the DRBG using a single internal entropy source with at least $2^{112}$ bits of entropy to provide random bits required for performing security functions such as cryptographic key generation.

- **Identification and authentication (FIA)**

  The SSO server performs identification and authentication based on the administrator ID during administrator authentication and requires administrator authentication prior to all operations. Additionally, it provides authentication feedback protection during authentication information input and blocks access for 5 minutes when five consecutive authentication failures occur. Furthermore, the SSO server uses a CSRF token to prevent authentication information reuse attempts when administrators log in.

  The SSO agent performs identification and authentication through initial authentication and authentication token-based authentication for general users and requires authentication prior to all operations. Additionally, it provides authentication feedback protection during authentication information input and blocks access for 5 minutes when five consecutive authentication failures occur. Furthermore, the SSO agent uses a CSRF token to prevent authentication information reuse attempts when general users log in. The SSO agent and the

SSO server perform mutual authentication through a proprietary protocol.

When issuing an authentication token, the SSO server generates the authentication token using a validated cryptographic module, and authentication token verification is also performed through the same validated cryptographic module.

To ensure account security for administrators and general users, password combination rules are enforced. Passwords must be at least 9 characters long and no more than 255 characters. They must contain at least one digit, one uppercase letter, one lowercase letter, and one special character. Furthermore, to enhance security, the following password settings are prohibited: Setting a password identical to the user's account ID is prohibited. Passwords consisting of consecutive repetitions of the same character or number are not permitted.
Furthermore, passwords formed by sequentially entering characters or numbers that are adjacent on the keyboard layout are prohibited. Additionally, reuse of the immediately preceding password is not allowed.
Detailed rules are as follows in the table below.

**[Table 12] Password Composition Rules for Administrator and General User Accounts**

| Category | Description |
|---|---|
| **Requirements** | Minimum length: 9 characters or more |
| | Maximum length: 255 characters or fewer |
| | The password includes at least one numeric character, one uppercase letter, one lowercase letter, and one special character |
| **Prohibited Items** | Passwords identical to the user account (ID) are prohibited |
| | Passwords containing consecutively repeated identical characters or numbers are prohibited |
| | Passwords containing sequential characters or numbers based on the keyboard layout are prohibited |
| | Reuse of the most recently used password is prohibited |

When generating an authentication token for general users during the single sign-on process, the authentication token is generated using a validated cryptographic module based on authentication token generation information. Additionally, upon authentication token destruction, the data is overwritten with '0' three times for secure destruction.

The TSF displays the password entered by administrators and general users as "▪" instead of the actual characters during the authentication process.
The TSF successfully identifies each user before performing any actions on behalf of that user.

- **Security management (FMT)**

The SSO server restricts security management functions such as general user management, administrator information management, audit data inquiry, access control policy management, and business system configuration settings so that they are provided only to authorized administrators. Authorized administrators can perform such management functions only through the security management interface.

TSF data management functions are permitted only to authorized administrators. The TSF enforces password change upon the administrator's initial access and provides administrator information modification and general user password generation functions only to authorized administrators.

The TSF classifies roles into authorized administrators as the super administrator role and general users as the user role. The super administrator has all privileges (Read/Write), and the general user can perform only the function of changing their own password.

Authorized administrators consist solely of super administrators, and the super administrator performs all security management functions of the TOE through the security management interface.

When an authorized administrator accesses the security management interface for the first time, password change is mandatorily required.

- **Protection of the TSF (FPT)**

  The SSO server applies the TLS v1.3 encrypted communication protocol based on JRE to ensure confidentiality and integrity when communicating with the SSO agent.
  For TSF data protection, authentication information of general users and administrators is encrypted for storage and management, and integrity verification information is also encrypted and managed. All critical data is stored and managed in encrypted form in files and the DBMS.
  In addition, TSF self-tests, TSF integrity tests, and cryptographic module self-tests are performed during initial boot and periodic operation, and TSF integrity tests are performed upon the administrator's manual request to ensure TSF data protection.

  The SSO agent applies the TLS v1.3 encrypted communication protocol based on JRE to ensure confidentiality and integrity when communicating with the SSO server.
  For TSF data protection, authentication information of general users is encrypted for storage and management, and integrity verification information is encrypted and managed. All critical data is stored and managed in encrypted form in files.
  In addition, TSF self-tests, TSF integrity tests, and cryptographic module self-tests are performed during initial boot and periodic operation to ensure TSF data protection.

The TSF protects important information stored in repositories controlled by the TSF, such as passwords, cryptographic keys, account information, authentication keys, and TOE configuration values, from unauthorized disclosure.

If entropy source errors such as noise source health test failure occur, the SSO server and SSO agent transition to a fatal error state, and in such cases, the validated cryptographic module and TOE operation are blocked. Thereafter, authorized administrators maintain a secure state by reinstalling the system and restarting the WAS server in accordance with the manual recovery procedures specified in the preparation procedures document.

- **TOE access (FTA)**

  When executing security management functions of the SSO server, concurrent sessions are restricted, and the maximum number of concurrent management access sessions belonging to the same administrator is limited to '1'. If an authorized administrator is already logged in and the same administrator account attempts to log in, the new access is blocked.

  For general user access sessions of the SSO agent, the maximum number of concurrent sessions is also limited to '1'.
  In addition, if an administrator session or general user session exceeds the designated inactivity timeout (fixed value: 10 minutes), the session is automatically terminated.
  Authorized administrators are restricted according to allowed IP access rules (maximum of 2 IP addresses), and the results of session restrictions are generated as audit data in the security management interface.

- **Trusted path/channels(FTP)**

  TOE provides a secure channel based on TLS v1.3, a standardized secure communication protocol, to ensure secure communication between external IT entities, TSFs.
  To protect information from security threats that may occur during data transmission,
  the TLS_AES_128_GCM_SHA256 cipher suite is applied, and the detailed security mechanisms are as follows.:

  - Key exchange method: Adopts the ECDHE (Elliptic Curve Diffie-Hellman Ephemeral) method, which generates a new key at each transmission point. Specifically, it uses the x25519 elliptic curve algorithm to provide high-security key exchange and perfect forward secrecy.

## 1.5. Conventions

This Security Target uses certain abbreviations and a mixture of English terms for clarity of meaning. The notation, format, and writing conventions used follow the Common Criteria.

The Common Criteria allows iteration, assignment, selection, and refinement operations to be performed in Security Functional Requirements. Each operation is used in this Security Target.

**Iteration**

This is used when a component is repeated multiple times by applying an operation in various ways. The result of an iteration operation is indicated by placing the iteration number in parentheses after the component identifier, that is, (iteration number).

**Assignment**

This is used to assign specific values to unspecified parameters (e.g., password length). The result of an assignment operation is indicated in square brackets, that is, [ assignment value ].

**Selection**

This is used to select one or more options provided by the Common Criteria for Information Technology Security Evaluation when describing requirements. The result of a selection operation is indicated in *underlined italics*.

**Refinement**

This is used to further restrict a requirement by adding detailed information. The result of a refinement operation is indicated in **bold**.

## 1.6. Terms and definitions

Terms used in this Security Target that are identical to those used in the Common Criteria shall follow the Common Criteria.

**Private Key**

A cryptographic key used in conjunction with an asymmetric cryptographic algorithm and uniquely associated with a single entity (the subject using the private key). It shall not be disclosed

**Object**

A passive entity within the TOE that contains or receives information and is the target of operations by a subject

**Health test**

Implemented within a random number generator to monitor the noise source in real time. A health test is not intended to verify statistical weaknesses of the noise source but to detect malfunction of the collected noise source due to equipment aging or similar causes

※ For detailed information, refer to the health tests defined in Clause 5.2 of TTAK.KO-12.0306/R1

**Deterministic Random Bit Generator (DRBG)**

An algorithm that generates a bit sequence from an initial value called a seed, producing the same bit sequence when the same seed is input

**Approved mode of operation**

The mode of operation of a cryptographic module that uses approved cryptographic algorithms

**Approved cryptographic algorithm**

A cryptographic algorithm selected by the cryptographic module validation authority, considering security, reliability, and interoperability, for block ciphers, hash functions, message authentication codes, random number generators, public key cryptography, digital signature algorithms, and related cryptographic algorithms

**Validated Cryptographic Module**

A cryptographic module that has been verified and approved by the cryptographic module validation authority and assigned a validation number

**Public Key**

A cryptographic key used in conjunction with an asymmetric cryptographic algorithm and uniquely associated with a single entity (the subject using the public key). It may be disclosed.

**Public Key(asymmetric) cryptographic algorithm**

A cryptographic algorithm that uses a public and private key pair

**Attack potential**

The degree of effort required to exploit a vulnerability in the TOE

Item note 1: Effort is expressed as a function of attributes related to the attacker (e.g., expertise, resources, and motivation) and attributes related to the vulnerability itself (e.g., window of opportunity, exposure time)

**Management access**

An act of an administrator attempting to access the TOE for management purposes using

HTTPS, SSH, TLS, etc

**Recommend / be recommended**

The terms "recommend" or "be recommended" in application notes do not mandate mandatory application to the TOE but indicate requirements recommended for safe operation of the TOE

**Random bit generator (RBG)**

A device or algorithm that outputs statistically independent and unbiased binary sequences. A random number generator used for cryptographic applications generally produces bit sequences of 0 and 1, which may be combined into random blocks. Random number generators are classified as deterministic or non-deterministic. A deterministic random number generator consists of an algorithm that generates a bit sequence from an initial value called a seed key, while a non-deterministic random number generator produces output dependent on unpredictable physical sources

※ A cryptographic random number generator consists of an entropy source used to construct the seed and a Deterministic Random Bit Generator (DRBG)

**Symmetric cryptographic technique**

A cryptographic technique that uses the same secret key in both encryption and decryption modes, also referred to as secret key cryptography

**Iteration**

The use of the same component to express two or more different requirements

**Security Target (ST)**

A specification of implementation-dependent security requirements for the TOE based on the security problem definition

**Security Policy Document**

A document published together with the cryptographic module name in the list of validated cryptographic modules, summarizing and specifying the cryptographic module form, approved cryptographic algorithms provided by the module, and its operational environment

**Protection Profile (PP)**

An implementation-independent specification of security requirements for a TOE type

**Decryption**

The process of restoring ciphertext to its original plaintext using a decryption key

**Secret Key**

A cryptographic key used in conjunction with a secret key cryptographic algorithm and uniquely associated with one or more entities. It shall not be disclosed

**User**

A human technical system or one of its components that interacts with the TOE from outside the TOE boundary. In the TOE, users include authorized administrators and authorized general users

※ User types related to SFR are classified as human user and external IT entity. Human users are further classified as local human users who directly interact with the TOE through TOE equipment and remote human users who indirectly interact with the TOE through other IT products.

**Selection**

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

**Seed**

A secret value used to initialize a random number generator

**Manual recovery**

Recovery performed by user execution or through an update server involving user intervention

**Identity**

A unique representation that identifies an authorized user. It may be the user's real name, abbreviation, or pseudonym.

**Encryption**

The process of transforming plaintext into ciphertext using an encryption key

KCMVP (Korea Cryptographic Module Validation Program)

A scheme that verifies the security and implementation conformance of cryptographic modules used to protect non-classified but important information transmitted through national and public institution information and communication networks

**Entropy**

A measure used to evaluate the unpredictability of data. It quantitatively represents the amount of information contained in data. It indicates disorder or randomness, and entropy increases as data becomes closer to true randomness.

**Entropy rate**

A value obtained by dividing the entropy of data by the size of the data, expressed between 0 and 1

**Entropy source**

A function or device combining a noise source, health tests, and a conditioning algorithm

**Business System**

An application server that authorized general users intend to access through 'Single Sign-On'

**Element**

An independent statement of security requirements allocated to SAR or SFR

**Role**

A predefined set of rules that establishes the interactions allowed between a user and the TOE

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

Modification or iteration of a component. The operations permitted on a component are assignment, iteration, refinement, and selection.

**Operation (on an object)**

A specific type of action performed by a subject on an object

**External Entity**

A human technical system or one of its components that interacts with the TOE from

outside the TOE boundary

**Threat Agent**

An entity that has the potential to carry out malicious actions against assets protected by the TOE

**Authorized Administrator**

An authorized user responsible for securely operating and managing the TOE

**Authorized User**

An entity that can perform operations on the TOE in accordance with the SFR

**End User**

A user who is not an authorized administrator of the TOE but intends to use the business system

**Authentication Data**

Information used to prove the identity of a user

**Authentication token**

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

**Self-Test**

Pre-operational and conditional tests executed by a validated cryptographic module

**Assets**

Entities to which the owner of the TOE assigns value

**Noise Source**

A function or device that generates non-deterministic data

**Refinement**

Specifying additional details in a security component

**Dependency**

A relationship between components in which, if a PP, ST, functional package, or assurance

package includes one component, it must also include other components on which it depends or provide a rationale if the dependent components are not included

**Subject**

An entity within the TOE that performs operations on objects

**Augmentation**

Adding one or more requirements to a package

Item note 1: For functional packages, such augmentation is considered only within a single package and does not need to be considered for other packages, PP, or ST.

Item note 2: For assurance packages, augmentation applies to one or more SARs.

**Conditioning**

The process of removing bias from collected noise sources to increase the entropy rate per bit

**Component**

The smallest selectable unit that forms the basis of requirements, consisting of a set of elements

**Class**

A collection of Common Criteria families that share the same security objective

**Family**

A collection of components that have similar purposes but differ in emphasis or rigor

**Target of Evaluation (TOE)**

A set of software, firmware, and/or hardware, possibly accompanied by documentation, that is the subject of evaluation

**Evaluation Assurance Level (EAL)**

A structured package of assurance requirements representing a predefined assurance level

Item note 1: EAL is defined in Part 5 of the CC.

**Can / could**

The terms “can” or “could” in application notes indicate requirements that may be applied

to the TOE at the discretion of the Security Target author

**Assignment**

Specifying parameters identified within a functional or assurance component

**Shall / must**

The terms "shall" or "must" in application notes indicate requirements that must be mandatorily applied to the TOE

**Critical Security Parameters (CSP)**

Security-related information that could compromise the security of a cryptographic module if disclosed or modified (e.g., secret keys/private keys, authentication data such as passwords or personal identification numbers)

**API (Application Programming Interface)**

A set of software libraries that exist between the application layer and the platform system layer to facilitate application development executed on the platform

**DBMS (Database Management System)**

A software system designed to construct and manage databases and support their application

**SSL (Secure Sockets Layer)**

A security protocol proposed by Netscape to provide security properties such as confidentiality and integrity over computer networks

**TLS (Transport Layer Security)**

An encrypted authentication communication protocol between server and client based on SSL, described in RFC 2246

**TOE Security Functionality (TSF)**

The combined functionality of all hardware, software, and firmware of the TOE that is relied upon for the enforcement of the SFR

**TSF Data**

Data created by the TOE for the TOE that may affect the operation of the TOE

**Wrapper**

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

# 2. Conformance

## 2.1. Conformance Claims

### 2.1.1. Common Criteria Conformance claim

The Common Criteria, Protection Profile, and security requirements package with which this Security Target and the TOE comply are as follows.

**[Table 13] Common Criteria Conformance Claim**

| | | |
|---|---|---|
| **Common Criteria** | | Common Criteria for Information Technology Security Evaluation CC:2022 Release 1<br>- Part 1: Introduction and General Model, CC:2022 r1 (CCMB-2022-11-001, November 2022)<br>- Part 2: Security Functional Components, CC:2022 r1 (CCMB-2022-11-002, November 2022)<br>- Part 3: Security Assurance Components, CC:2022 r1 (CCMB-2022-11-003, November 2022)<br>- Part 4: Framework for Evaluation Methodology and Activities, CC:2022 r1 (CCMB-2022-11-004, November 2022)<br>- Part 5: Pre-defined Packages of Security Requirements,<br>- Common Criteria Evaluation Methodology for Information Security Systems, CCMB-2022-11-006, CEM:2022 Revision 1, November 2022.<br>- Common Criteria for Information Security Systems CC:2022 R1 Errata and Interpretation for CC:2022 (Release 1) and CEM:2022 (Release 1), Version 1.2, CCMB-2025-001, October 2025. |
| **PP (Protection Profile)** | | Conformant to the National Integrated Authentication Protection Profile v3.1 (2024-06-27) |
| **Conformance Type** | **Part 2 Security Functional Components** | Part 2 Extended:<br>FIA_IMA.1, FIA_SOS.3, FMT_PWD.1, FPT_PST.1 |
| | **Part 3** | Part 3 Conformant |

| | | |
|---|---|---|
| | **Security Assurance Components** | |
| | **Package** | Augmented: EAL1 augmented with ATE_FUN.1 |

### 2.1.2. Conformance Type

This Security Target claims 'strict Protection Profile conformance'.

### 2.1.3. Protection Profile Composition Conformance Claim

The Protection Profile with which this Security Target complies is 'National Single Sign-On Protection Profile V3.1', and there are no other Protection Profiles that are composed.

### 2.1.4. PP Conformance Claim

The Protection Profile with which this Security Target complies is 'National Single Sign-On Protection Profile V3.1'.
In addition, this Security Target contains only a PP conformance claim, and as it is based on CC:2022, it includes only a direct conformance claim to the referenced PP.

### 2.1.5. Package conformance claim

The assurance requirements package with which this Security Target complies is EAL1, and certain assurance requirements are additionally defined.
- Assurance package: EAL1 augmented (ATE_FUN.1)

### 2.1.6. Conformance claim rationale

This Security Target adopts the TOE type, security problem definition, security objectives, and security requirements of the Protection Profile without modification; therefore, the conformance claim to 'National Single Sign-On Protection Profile V3.1' is 'strict Protection Profile conformance'.

## 2.2. Method of Compliance

### 2.2.1. Reference to evaluator and developer action elements

The package with which this Security Target complies requires the use of the evaluation methods/evaluation activities defined in <6.2 Assurance Requirements>.

※ The evaluation methods/evaluation activities defined in <6.2 Assurance Requirements> are written to be performed including newly added or modified work units in CEM:2022.

# 3. Security problem definition

The security problem definition defines the threats, organizational security policies, and assumptions intended to be addressed by the TOE and its operational environment.

## 3.1. Assets

The primary assets protected by the single sign-on are as follows.

- Internal IT resources and services integrated with the single sign-on
- The TOE itself and important data related to TOE operation (e.g., TSF data)

## 3.2. Threats

A threat agent is an IT entity or user that attempts unauthorized access to protected assets or causes harm through abnormal methods and may generate various threats as follows. In this case, the threat agent to the TOE is assumed to have basic levels of expertise, resources, and motivation.

### 3.2.1. Unauthorized access

**T.SESSION_HIJACK**

A threat agent may gain user privileges by accessing a user screen left logged in or by using a user session that has not been terminated after logout.

**T.RETRY_AUTH_ATTEMPT**

A threat agent may repeatedly attempt authentication and use the information obtained to successfully authenticate and then access the TOE by impersonating an authorized user.

**T.IMPERSONATION**

A threat agent may impersonate an authorized user or TOE component to gain access to the TOE.

**T.REPLAY**

A threat agent may obtain and copy authentication information and reuse it to access the TOE.

**T.WEAK_PASSWORD**

A threat agent may obtain poorly managed passwords, such as default passwords, and impersonate an authorized user to access the TOE. If weak password rules are applied, the threat agent may also impersonate an authorized user to access the TOE.

### 3.2.2. Information disclosure

**T.STORED_DATA_LEAKAGE**

A threat agent may disclose important data (e.g., cryptographic keys, TOE configuration values) stored within the TOE or in external entities interacting with the TOE (e.g., DBMS) through unauthorized means.

**T.TRANSMISSION_DATA_DAMAGE**

A threat agent may disclose or modify transmission data between TOE components or between the TOE and external IT entities through unauthorized means.

**T.WEAK_CRYPTO_PROTOCOLS**

A threat agent may analyze traffic using weak cryptographic communication protocols or low cryptographic strength to infer cryptographic key information or determine the contents of encrypted communications.

### 3.2.3. Compromise of the TOE security functionality

**T.TSF_COMPROMISE**

A threat agent may compromise the TSF through unauthorized access or other means, causing malfunction of TOE functions or disabling TOE functionality.

## 3.3. Organizational security policies

**P.AUDIT**

Security-related events shall be recorded and maintained in order to ensure accountability for security-related actions, and the recorded data shall be reviewed. In addition, the available space of the audit data storage disk shall be periodically checked to prevent loss of audit data, and stored audit data shall be protected from unauthorized modification and deletion.

**P.SECURE_OPERATION**

Management measures shall be provided to ensure that administrators securely configure the TOE in compliance with the organization's single sign-on security policy and operate the TOE correctly in accordance with the TOE operation manual.

**P.CRYPTO_STRENGTH**

The organization shall apply encryption measures to the storage and transmission of important data, such as passwords used for user authentication, and shall use secure cryptographic algorithms.

## 3.4. Assumptions

The following assumptions are made about the TOE operational environment.

**A.PHYSICAL_CONTROL**

The locations where the SSO agent and SSO server, as TOE components, are installed and operated shall be equipped with access control and protective facilities to allow access only to authorized administrators.

**A.MANUAL_RECOVERY**

Procedural manual recovery, such as reinstallation with user intervention, shall be supported to restore altered information after failure or service interruption of the SSO agent and SSO server, as TOE components.

**A.TRUSTED_ADMIN**

Authorized administrators of the TOE are assumed to be non-malicious, properly trained in TOE management functions, and to perform their duties accurately in accordance with administrative guidance.

**A.TRUSTED_TIMESTAMP**

It is assumed that the TOE uses a reliable timestamp provided by the operational environment to accurately record security-related events.

**A.SECURE_DBMS**

The DBMS shall be installed on the same operating system as the TOE and shall use its identification and authentication functions to protect against deletion or modification by unauthorized users.

**A.SECURE_DEVELOPMENT**

Developers integrating user identification and authentication functions into the business system operational environment using the TOE shall comply with the requirements specified in the documentation provided with the TOE to ensure secure application of the TOE security functions.

**A.SECURED_ADMIN_ACCESS**

The web server of the management server operational environment and the web browser of the administrator PC shall communicate using a secure channel.

**A.OPERATION_SYSTEM_REINFORCEMEN**

The operating system on which the TOE is installed and operated shall be reinforced against the latest vulnerabilities to ensure its reliability and security.

# 4. Security objectives

This Security Target defines security objectives by classifying them into security objectives for the TOE and security objectives for the operational environment. Security objectives for the TOE are those directly addressed by the TOE, while security objectives for the operational environment are those that shall be addressed by technical or procedural measures supported by the operational environment to ensure that the TOE correctly provides its security functionality.

## 4.1. Security objectives for the operational environment

The following are security objectives that shall be addressed by technical or procedural measures supported by the operational environment to ensure that the TOE correctly provides its security functionality.

**OE.LOG_BACKUP**

Authorized administrators of the TOE shall periodically check the available space of the audit data repository to prevent loss of audit records and shall perform audit record backup (e.g., external log server, separate storage device) to ensure that audit records are not exhausted.

**OE.PHYSICAL_CONTROL**

The TOE components, SSO server and SSO agent, shall be located in a physically secure environment accessible only to authorized administrators.

**OE.MANUAL_RECOVERY**

Procedural manual recovery, such as reinstallation with user intervention, shall be supported to restore altered information after failure or service interruption of the SSO agent and SSO server, as TOE components.

**OE.TRUSTED_ADMIN**

Authorized administrators of the TOE are assumed to be non-malicious, properly trained in TOE management functions, and to perform their duties accurately in accordance with all administrative guidelines and procedures.

**OE.TRUSTED_TIMESTAMP**

The TOE shall use a reliable timestamp provided by the operational environment to accurately record security-related events.

**OE.SECURE_DBMS**

The DBMS shall securely store and protect audit data and TSF data generated by the TOE.

**OE.SECURE_DEVELOPMENT**

Developers integrating user identification and authentication functions into the business system

operational environment using the TOE shall comply with the requirements specified in the documentation provided with the TOE to ensure secure application of the TOE security functions.

**OE.SECURED_ADMIN_ACCESS**

The confidentiality and integrity of data transmitted between the administrator PC web browser and the web server in the management server operational environment shall be ensured.

**OE.OPERATION_SYSTEM_REINFORCEMEN**

The operating system on which the TOE is installed and operated shall be reinforced against the latest vulnerabilities to ensure its reliability and security.

## 4.2. Security objectives rationale

### 4.2.1. Rationale for the security objectives for the operational environment

The rationale for the security objectives demonstrates that the specified security objectives are appropriate and sufficiently address the security problem as follows.

- Each threat, organizational security policy, and assumption is addressed by at least one security objective.
- Each security objective addresses at least one threat, organizational security policy, or assumption.
- Since assumptions are always set for the TOE operational environment, TOE security objectives are not considered assumptions.

**[Table 14] Mapping of Security Objectives for the Operational Environment**

| | OE.LOG_BACKUP | OE.PHYSICAL_CONTROL | OE.MANUAL_RECOVERY | OE.TRUSTED_ADMIN | OE.TRUSTED_TIMESTAMP | OE.SECURE_DBMS | OE.SECURE_DEVELOPMENT | OE.SECURED_ADMIN_ACCESS | OE.OPERATION_SYSTEM_REINFORCEMEN |
|---|---|---|---|---|---|---|---|---|---|
| P.AUDIT | X | | | | | | | | |
| P.SECURE OPERATION | | | | X | | | | | |
| A.PHYSICAL_CONTROL | | X | | | | | | | |
| A.MANUAL_RECOVERY | | | X | | | | | | |
| A.TRUSTED_ADMIN | X | | | X | | | | | |
| A.TRUSTED_TIMESTAMP | | | | | X | | | | |
| A.SECURE_DBMS | | | | | | X | | | |
| A.SECURE_DEVELOPMENT | | | | | | | X | | |
| A.SECURED_ADMIN_ACCESS | | | | | | | | X | |
| A.OPERATION_SYSTEM_REINFORCEMEN | | | | | | | | | X |

| **P.AUDIT** | **OE.LOG BACKUP** |
|---|---|

**P.AUDIT** is fulfilled by **OE.LOG_BACKUP**.

**OE.LOG_BACKUP** ensures that, in addition to the TOE functions, periodic inspection of the audit data

storage space is performed by the administrator and that regular log backup or log transmission to an external log server is carried out to prevent loss of audit records.

| | |
|---|---|
| **P.SECURE_OPERATION** | **OE.TRUSTED_ADMIN** |

**P.SECURE_OPERATION** is achieved by **OE.TRUSTED_ADMIN**.

**OE.TRUSTED_ADMIN** ensures that administrators operate the TOE correctly in accordance with the organization's single sign-on security policy and operation manual.

| | |
|---|---|
| **A.PHYSICAL_CONTROL** | **OE.PHYSICAL_CONTROL** |

**A.PHYSICAL_CONTROL** is supported by **OE.PHYSICAL_CONTROL.**

**OE.PHYSICAL_CONTROL** ensures that the SSO server and SSO agent are installed in locations equipped with protective facilities and that access is controlled to allow only authorized administrators.

| | |
|---|---|
| **A.MANUAL_RECOVERY** | **OE.MANUAL_RECOVERY** |

**A.MANUAL_RECOVERY** is achieved by **OE.MANUAL_RECOVERY.**

**OE.MANUAL_RECOVERY** ensures that the TOE agent/server documentation specifies procedural manual recovery methods such as TOE reinstallation, enabling administrators to restore tampered information (configuration values, libraries).

| | |
|---|---|
| **A.TRUSTED_ADMIN** | **OE.LOG_BACKUP, OE.TRUSTED_ADMIN** |

**A.TRUSTED_ADMIN** is supported by **OE.TRUSTED_ADMIN** and **OE.LOG_BACKUP**.

**OE.TRUSTED_ADMIN** ensures that administrators are non-malicious, properly trained in TOE management functions, and perform their duties accurately in accordance with administrative guidance.

**OE.LOG_BACKUP** ensures that authorized administrators periodically check the available space of the audit data repository and perform audit record backup (e.g., external log server, separate storage device) to prevent loss of audit records.

| | |
|---|---|
| **A.TRUSTED_TIMESTAMP** | **OE.TRUSTED_TIMESTAMP** |

**A.TRUSTED_TIMESTAMP** is supported by **OE.TRUSTED_TIMESTAMP.**

**OE.TRUSTED_TIMESTAMP** ensures that a reliable timestamp provided by the operational environment is used to accurately record security-related events.

| | |
|---|---|
| **A.SECURE_DBMS** | **OE.SECURE_DBMS** |

**A.SECURE_DBMS** is supported by **OE.SECURE_DBMS.**

**OE.SECURE_DBMS** ensures that audit records are stored in a DBMS installed on the same operating system as the TOE and that the DBMS identification and authentication functions are used to protect against deletion or modification by unauthorized users.

| | |
|---|---|
| **A.SECURE_DEVELOPMENT** | **OE.SECURE_DEVELOPMENT** |

**A.SECURE_DEVELOPMENT** is supported by **OE.SECURE_DEVELOPMENT.**

**OE.SECURE_DEVELOPMENT** ensures that developers integrating user identification and authentication functions into the business system operational environment comply with the requirements specified in the documentation provided with the TOE to securely apply the TOE security functions.

| | |
|---|---|
| **A.SECURED_ADMIN_ACCESS** | **OE.SECURED_ADMIN_ACCESS** |

**A.SECURED_ADMIN_ACCESS** is achieved by **OE.SECURED_ADMIN_ACCESS.**

**OE.SECURED_ADMIN_ACCESS** ensures the confidentiality and integrity of communication between the administrator PC web browser and the web server using a secure channel.

| | |
|---|---|
| **A.OPERATION_SYSTEM_REINFORCEMEN** | **OE.OPERATION_SYSTEM_REINFORCEMEN** |

**A.OPERATION_SYSTEM_REINFORCEMEN** is supported by **OE.OPERATION_SYSTEM_REINFORCEMEN.**

**OE.OPERATION_SYSTEM_REINFORCEMEN** ensures the reliability and security of the operating system by applying patches for the latest vulnerabilities in the operating system on which the TOE is installed and operated.

# 5. Extended components definition

## 5.1. Identification and authentication(FIA)

### 5.1.1. 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) Minimal: Success and failure of mutual authentication

#### 5.1.1.1. FIA_IMA.1 TOE Internal mutual authentication

Hierarchical to : No other components.

Dependencies : No dependencies.

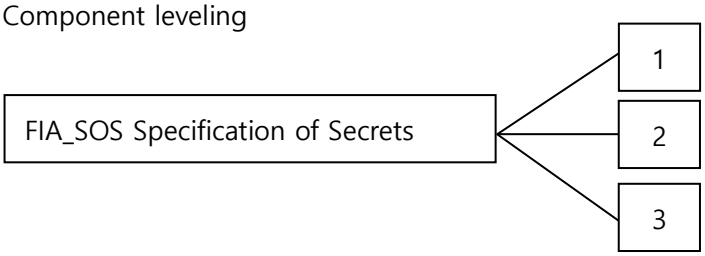
FIA_IMA.1.1 The TSF shall perform mutual authentication between [assignment: different parts of TOE] using the [assignment: authentication protocol] that meets the following [assignment: *list of standards*]

### 5.1.2. Specification of Secrets

Family Behaviour

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

Component leveling




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

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

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

Management : FIA_SOS.3

There are no management activities foreseen.

Audit : FIA_SOS.3

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

a) Minimal : Success and failure of the activity

#### 5.1.2.1. 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*].

## 5.2. Security management(FMT)

### 5.2.1. 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) Minimal: All changes of the password

#### 5.2.1.1. 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*].

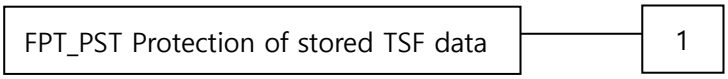
## 5.3. Protection of the TSF(FPT)

### 5.3.1. Protection of stored TSF data

Family Behaviour

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

Component leveling




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

Management : FPT_PST.1

There are no management activities foreseen.

Audit : FPT_PST.1

There are no auditable events foreseen.

#### 5.3.1.1. 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*].

# 6. Security requirements

This chapter specifies the security functional requirements and assurance requirements that shall be satisfied by the TOE.

## 6.1. Security functional requirements

The security functional requirements defined in this Security Target are derived from the relevant security functional components of CC Part 2 in order to satisfy the security objectives identified in Chapter 4. The security functional components used are summarized in the table below.

**[Table 15] Security Function Component Summary**

| Security functional class | Security functional component | |
|---|---|---|
| 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.1 | Audit data storage location |
| | FAU_STG.4 | Action in case of possible audit data loss |
| | FAU_STG.5 | Prevention of audit data loss |
| Cryptographic support (FCS) | FCS_CKM.1 | Cryptographic key generation |
| | FCS_CKM.2 | Cryptographic key distribution |
| | FCS_CKM.5 | Cryptographic Key Derivation |
| | FCS_CKM.6 | Timing and event of cryptographic key destruction |
| | FCS_COP.1 | Cryptographic operation |
| | FCS_RBG.1 | Random bit generation (RBG) |
| | FCS_RBG.3 | Random bit generation (Internal Seeding - Single Source) |
| Identification and authentication (FIA) | FIA_AFL.1(1) | Authentication failure handling (Administrator) |
| | FIA_AFL.1(2) | Authentication failure handling (General User) |
| | FIA_IMA.1 | TOE Internal mutual authentication (Extended) |
| | FIA_SOS.1 | Verification of secrets |
| | FIA_SOS.2 | TSF Generation of secrets |
| | FIA_SOS.3 | Destruction of secrets (Extended) |
| | FIA_UAU.2 | Timing of authentication |
| | FIA_UAU.4(1) | Single-use Authentication Mechanisms (Administrator) |
| | FIA_UAU.4(2) | Single-use Authentication Mechanisms (General User) |

| | | |
|---|---|---|
| | FIA_UAU.7 | Protected authentication feedback |
| | FIA_UID.2 | Timing of identification |
| Security management (FMT) | FMT_MOF.1 | Security Function Management |
| | FMT_MTD.1 | Management of TSF data |
| | FMT_PWD.1 | Management of ID and password(Extended) |
| | FMT_SMF.1 | Specification of Management Functions |
| | FMT_SMR.1 | Security Role |
| Protection of the TSF (FPT) | FPT_FLS.1 | Maintenance of a secure state upon failure |
| | FPT_ITT.1 | Basic internal TSF data transfer protection |
| | FPT_PST.1 | Basic protection of stored TSF data (Extended) |
| | FPT_TST.1 | TSF testing |
| TOE access (FTA) | FTA_MCS.2 | Per user attribute limitation on multiple concurrent sessions |
| | FTA_SSL.3 | TSF-initiated termination |
| | FTA_TSE.1 | TOE session establishment |
| Trusted path/channels (FTP) | FTP_ITC.1 | Inter-TSF trusted channel |

### 6.1.1. Security audit (FAU)

#### 6.1.1.1. 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 [ [Table 16] Security violation Action List ] upon detection of a potential security violation.

**[Table 16] Security violation Action List**

| Security violation | Response action |
|---|---|
| When administrator authentication attempts fail consecutively for the defined number of times (fixed value: 5) | - Disable the authentication function for a defined period (fixed value: 5 minutes)<br>- Send a warning message email to the authorized administrator |
| When general user authentication attempts fail consecutively for the defined number of times (fixed | - Disable the authentication function for a defined period (fixed value: 5 minutes)<br>- Send a warning message email to the authorized |

| | |
|---|---|
| value: 5) | administrator |
| When the SSO server startup or periodic self-test fails | - Terminate the server process<br>- Send a warning message email to the authorized administrator |
| When the SSO agent startup or periodic self-test fails | - Send a warning message email to the authorized administrator |
| When an authorized administrator manually initiates an integrity test and it fails | - Send a warning message email to the authorized administrator |
| When the audit evidence storage capacity exceeds the threshold (fixed value: 80%) | - Send a warning message email to the authorized administrator |
| When the audit evidence storage reaches a full state (fixed value: 90%) | - Send a warning message email to the authorized administrator<br>- After recording the audit event indicating the full state, no further audit records shall be accepted<br>- Ignore the audited event |

#### 6.1.1.2. 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) [ [Table 17] 'Auditable Events' of Auditable Events ]

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, [ [Table 17] "Additional Audit Record Content" for Auditable Events ]

**[Table 17] Auditable Events**

| Sub-category | Audit events | Additional audit information |
|---|---|---|
| Security audit | Startup and shutdown of the SSO server and SSO agent | |
| | Response actions and results when audit storage fails | |
| Cryptographic support | Cryptographic key generation failure | |
| | Cryptographic operation failure (including the type of cryptographic operation) | |
| Identification and authentication | User login and logout | |
| | User registration, change and deletion | |
| | Actions taken when the limit on user authentication attempts is reached | |
| | All changes of the password | |
| | Success or failure of mutual authentication between the SSO server and SSO agent | |
| | SSO authentication token generation or verification results (success or failure) | |
| | SSO authentication token destruction and its result (success or failure) | |
| Security management | IP registration, deletion and change of administrative terminals | |
| | Execution of security management functions and all changes and deletions of security attribute values | Modified security attribute data |
| | Default account(ID)/Password change | |
| | Changes in agent registration status | |
| Protection of the TSF | Self-test of the SSO server and SSO agent | Failed security function |
| | Integrity verification of the SSO server and SSO agent | Component for which integrity verification has failed |
| TOE access | User session termination | |
| | Response action upon detection of duplicate login attempts using the same account | |
| | Rejection of new sessions based on concurrent | |

| | | |
|---|---|---|
| | session limits | |
| | Blocking of management terminal access IP addresses | |

### 6.1.1.3. 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 known [
When administrator authentication attempts fail consecutively for the defined number of times (fixed value: 5),
When general user authentication attempts fail consecutively for the defined number of times (fixed value: 5),
When a validated cryptographic module self-test fails within the SSO server,
When integrity verification fails within the SSO server,
When a process test fails within the SSO server,
When a validated cryptographic module self-test fails within the SSO agent,
When integrity verification fails within the SSO agent,
When a process test fails within the SSO agent,
When the audit data repository threshold (fixed value: 80%) is exceeded,
When the audit data repository reaches saturation (fixed value: 90%) ]
that indicate potential security violations

b) [ None ]

### 6.1.1.4. 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.

#### 6.1.1.5. 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 capability to apply [assignment: methods of [ descending order based on the event time and date ] of audit data based on [ event date AND event type ].

#### 6.1.1.6. FAU_STG.1 Audit data storage location

Hierarchical to : No other components.
Dependencies : FAU_GEN.1 Audit data generation.
FTP_ITC.1 Inter-TSF trusted channel
FAU_STG.1.1 The TSF shall be able to store generated audit data in [ *the local DBMS* ].

#### 6.1.1.7. FAU_STG.4 Action in case of possible audit data loss

Hierarchical to : No other components.
Dependencies : FAU_STG.2 Protection of the audit data repository

FAU_STG.4.1 The TSF shall perform [ send an email to the authorized administrator, [none] ] when the audit data repository exceeds [ the ratio of used storage space to total audit trail storage reaches the fixed threshold: 80% ].

#### 6.1.1.8. FAU_STG.5 Prevention of audit data loss

Hierarchical to : FAU_STG.4 Action in case of possible audit data loss.
Dependencies : FAU_STG.2 Protected audit trail storage

FAU_STG.5.1 The TSF *shall ignore audited events* and [ send a warning email to the authorized administrator ] if the audit trail is full.

## 6.1.2. Cryptographic support(FCS)

### 6.1.2.1. FCS_CKM.1 Cryptographic key generation

Hierarchical to : No other components.

Dependencies : [FCS_CKM.2 Cryptographic Key Distribution or
FCS_CKM.5 Cryptographic Key Derivation or
FCS_COP.1 Cryptographic operation]
[FCS_RBG.1 Random Bit Generation or
FCS_RNG.1 Generation of random numbers]
FCS_CKM.6 Timing and event of cryptographic key destruction

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with the specified cryptographic key generation algorithm [ [Table 18] "Cryptographic Algorithm" of the List of Cryptographic Key Generation Standards ] and specified cryptographic key sizes [ [Table 18] "Cryptographic Key Length" of the List of Cryptographic Key Generation Standards ] that meet the following: [ [Table 18] "Referenced Standard" of the List of Cryptographic Key Generation Standards ].

**[Table 18] List of Cryptographic Key Generation Standards**

| Purpose | Cryptographic Algorithm | Cryptographic Key Length | Referenced Standard |
|---|---|---|---|
| Integrity verification key | Hash_DRBG (SHA-384) | 256 bit | TTAK.KO-12.0331 |
| DEK (Data Encryption Key) | Hash_DRBG (SHA-384) | 256 bit | TTAK.KO-12.0331 |
| Authentication token encryption key | Hash_DRBG (SHA-384) | 128 bit | TTAK.KO-12.0331 |
| Session key for transmission data encryption | Hash_DRBG (SHA-384) | 128 bit | TTAK.KO-12.0331 |
| SSO server public/private key pair for encryption | RSAES | 3072 bit | ISO/IEC 18033-2 |
| SSO server public/private key pair for digital signature | RSA-PSS | 3072 bit | ISO/IEC 18033-2 |
| SSO agent public/private key pair for encryption | RSAES | 3072 bit | ISO/IEC 18033-2 |
| SSO agent public/private key pair for digital signature | RSA-PSS | 3072 bit | ISO/IEC 18033-2 |
| KEK (Key Encryption Key) | PBKDF2 (SHA-256) | 256 bit | TTAK.KO-12.0334-Part2 |

### 6.1.2.2. 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 [ [Table 19] "Cryptographic Key Distribution Method" of the List of Cryptographic Key Distribution Standards ] that meets the following: [ [Table 19] "Referenced Standard" of the List of Cryptographic Key Distribution Standards ].

**[Table 19] Cryptographic Key Distribution Standard List**

| Purpose | Cryptographic Algorithm | Cryptographic Key Length | Referenced Standard |
|---|---|---|---|
| Public Key Cryptography | RSAES (SHA-256) | Public Key 3072 bit | ISO/IEC 18033-2 |
| **Cryptographic Key Distribution Method** | | | |
| Use RSAES public key encryption provided by the validated cryptographic module. Securely encrypt and distribute the session key for encrypting transmission data between the SSO server and SSO agent using the RSA public key of the target server. | | | |
| **Purpose** | **Cryptographic Algorithm** | **Cryptographic Key Length** | **Referenced Standard** |
| TLS v1.3 Communication Session Key | ECDHE | 256 bit | rfc8446<br>rfc7748 |
| **Key Distribution Method** | | | |
| To establish a secure communication session between the SSO agent and the SSO server, session key exchange shall be performed using the Elliptic Curve Diffie-Hellman Ephemeral (ECDHE) algorithm. | | | |

### 6.1.2.3. FCS_CKM.5 Cryptographic Key derivation

Hierarchical to : No other components.

Dependencies : [FCS_CKM.2 Cryptographic key distribution or
FCS_COP.1 Cryptographic operation]
FCS_CKM.6 Timing and event of cryptographic key destruction

FCS_CKM.5.1 The TSF shall [Table 20] Cryptographic Key Derivation Standard List TSF conforms to the specified key derivation algorithm [ [Table 20] Cryptographic Key Derivation Standard List 'Reference Standards' ] and the specified cryptographic key length [ [Table 20] Cryptographic Key Derivation Standard List 'Cryptographic Key Length' ] must derive the cryptographic key [ KEK (Key Encryption Key) ] from [ password parameters directly entered by an authorized administrator ].

**[Table 20] Cryptographic Key Derivation Standard List**

| Purpose | TOE Module | Cryptographic Algorithm | Cryptographic Key Length | Referenced Standard |
|---|---|---|---|---|
| KEK(Key Encryption Key) | SSO Server | PBKDF2 (SHA-256) | 256 bit | TTAK.KO-12.0334-Part2 |
| KEK(Key Encryption Key) | SSO Agent | PBKDF2 (SHA-256) | 256 bit | TTAK.KO-12.0334-Part2 |
| TLS v1.3 Communication Session Key | SSO Agent<br>SSO Server | HKDF-SHA256 | 128 bit | rfc5869 |

### 6.1.2.4. FCS_CKM.6 Timing and event of cryptographic key destruction

Hierarchical to : No other components.

Dependencies : [FDP_ITC.1 User data flow without security attributes or
FDP_ITC.2 User Data Flow with Security Attributes or
FCS_CKM.1 Cryptographic Key Generation or
FCS_CKM.5 Cryptographic Key Derivation]

FCS_CKM.6.1 *When no longer required*, the TSF must destroy the cryptographic key list [ [Table 21] Cryptographic Key List ].

FCS_CKM.6.2 The TSF shall destroy cryptographic keys and keying material specified by FCS_CKM.6.1 in accordance with a specified cryptographic key destruction method [method of overwriting three times with the value '0'] that meets the following: [None].

**[Table 21] Cryptographic Key List**

| Purpose | Cryptographic Algorithm | Cryptographic Key Length | Destruction Period |
|---|---|---|---|
| KEK<br>(Key Encryption Key) | ARIA/CBC | 256 bit | Immediately after use (Memory) |
| Key for integrity verification | HMAC (SHA-256) | 256 bit | Upon invocation of |

| | | | |
|---|---|---|---|
| | | | product termination process (Memory) |
| DEK (Data Encryption Key) | ARIA/CBC | 256 bit | Upon invocation of product termination process (Memory) |
| Key for authentication token encryption | ARIA/CBC | 128 bit | Immediately after use (Memory) |
| Session key for transmission data encryption | ARIA/CBC | 128 bit | Immediately after use (Memory) |
| Private Key for Encryption (SSO Server) | RSAES (SHA-256) | 3072 bit | Upon invocation of product termination process (Memory) |
| Private Key for Digital Signature (SSO Server) | RSA-PSS (SHA-256) | 3072 bit | Upon invocation of product termination process (Memory) |
| Private Key for Encryption (SSO Agent) | RSAES (SHA-256) | 3072 bit | Upon invocation of product termination process (Memory) |
| Private Key for Digital Signature (SSO Agent) | RSA-PSS (SHA-256) | 3072 bit | Upon invocation of product termination process (Memory) |
| TLS v1.3 Communication Session Key | ECDHE | 256 bit | Immediately after use (Memory) |
| TLS v1.3 Communication Session Key | HKDF | 128 bit | Immediately after use (Memory) |
| TLS v1.3 Communication Session Key | AES_128_GCM | 128 bit | Immediately after use (Memory) |

#### 6.1.2.5. FCS_COP.1 Cryptographic operation

Hierarchical to : No other components.

Dependencies : [FDP_ITC.1 User data input without security attributes or
FDP_ITC.2 User data input with security attributes or
FCS_CKM.1 Cryptographic Key Generation or
FCS_CKM.5 Cryptographic Key Derivation]
FCS_CKM.6 Timing and event of cryptographic key destruction

FCS_COP.1.1 TSF shall perform the specified cryptographic operation [ [Table 22] Cryptographic Operations Standard List "cryptographic algorithm" ] according to the specified cryptographic key length [ [Table 22] Cryptographic Operations Standard List "cryptographic key length" ] and the specified cryptographic operation [ [Table 22] Cryptographic Operations Standard List "purpose" ].

**[Table 22] Cryptographic Operations Standard List**

| Purpose | Cryptographic Algorithm | Key Length | TOE Module | Referenced Standard |
|---|---|---|---|---|
| TOE Integrity Verification | HMAC (SHA-256) | 256 bit | SSO Server<br>SSO Agent | ISO/IEC 9797-2 |
| One-way Encryption of User Password | SHA-256 | 256 bit | SSO Server | ISO/IEC 10118-3 |
| DEK Encryption,<br>TSF Data Encryption | ARIA/CBC | 256 bit | SSO Server<br>SSO Agent | KS X 1213-1 |
| Authentication Token Encryption | ARIA/CBC | 128 bit | SSO Server | KS X 1213-1 |
| Encryption of Transmission Information Session Key | RSAES (SHA-256) | Public Key 3072 bit | SSO Server<br>SSO Agent | ISO/IEC 18033-2 |
| Authentication Token Digital Signature Generation/Verification, Mutual Authentication | RSA-PSS (SHA-256) | Public Key 3072 bit | SSO Server<br>SSO Agent | ISO/IEC 18033-2 |
| TLS v1.3 Secure Communication | TLS_AES_128_GCM (SHA256) | 128 bit | SSO Server<br>SSO Agent | rfc8446<br>ISO/IEC 18033-3 |

### 6.1.2.6. FCS_RBG.1 Random bit generation (RBG)

Hierarchical to : No other components.
Dependencies : [FCS_RBG.2 Random bit generation (External Seeding) or
FCS_RBG.3 Random Bit Generation (Internal Seeding - Single Source)]

FPT_FLS.1 Maintenance of a secure state upon failure
FPT_TST.1 TSF testing

FCS_RBG.1.1 The TSF shall perform deterministic random bit generation services using the [ Hash_DRBG (SHA-384) algorithm ] in accordance with [ TTAK.KO-12.0331 ] after initialization.

FCS_RBG.1.2 The TSF shall use the *TSF entropy source* *[ new SecureRandom().nextBytes() ]* for initialization and seeding.

FCS_RBG.1.3 The TSF shall update the DRBG state by *reseeding* using the *TSF entropy source* *[ new SecureRandom().nextBytes() ]* in accordance with [ TTAK.KO-12.0235/R2 ] under the following condition:
o The following situation
— *When the reseed counter reaches the limit of [ 1000 reseed operations ].*

#### 6.1.2.7. FCS_RBG.3 Random bit generation(Internal Seeding - Single Source)

Hierarchical to : No other components.
Dependencies : FCS_RBG.1 Random bit generation (Extended)(RBG)

FCS_RBG.3.1 TSF must have a minimum entropy of at least [ $2^{258}$ ] bits and must be able to seed the DRBG using the *TSF software-based entropy source* [ new SecureRandom().nextBytes() ].

### 6.1.3. Identification and authentication(FIA)

#### 6.1.3.1. FIA_AFL.1(1) Authentication failure handling (Administrator)

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

FIA_AFL.1.1 TSF shall detect when [ *5* ] failed authentication attempts related to [Administrator Authentication Attempt] occur.

FIA_AFL.1.2 When the defined number of failed authentication attempts *is reached*, the TSF shall perform [Administrator Account Lockout for 5 Minutes].

#### 6.1.3.2. FIA_AFL.1(2) Authentication failure handling (General Users)

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

FIA_AFL.1.1 TSF must detect when *[5]* failed authentication attempts related to [General User Authentication Attempts] occur.

FIA_AFL.1.2 When the number of failed authentication attempts *is reached* the defined threshold, the TSF shall [ lock the general user account for 5 minutes ].

#### 6.1.3.3. FIA_IMA.1 TOE Internal mutual authentication (Extended)

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

FIA_IMA.1.1 The TSF shall perform mutual authentication between the [ SSO Agent and SSO

Server ] using a [ proprietary protocol ] in accordance with [ none ].

#### 6.1.3.4. 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 secret information satisfies [ [Table 23] Password Security Criteria Type (1) ].

**[Table 23] Password Security Criteria Type (1)**

| Category | Description |
|---|---|
| Compliance Requirements | Minimum length: 9 characters or more |
| | Maximum length: 255 characters or fewer |
| | Must include at least one digit, one uppercase letter (A–Z), one lowercase letter (a–z), and one special character |
| Prohibited Items | Prohibited from setting the password identical to the user account (ID) |
| | Prohibited from consecutive repetition of identical characters or numbers |
| | Prohibited from sequential input of adjacent keyboard characters or numbers |
| | Prohibited from reusing the previously used password |

#### 6.1.3.5. FIA_SOS.2 TSF Generation of secrets

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

FIA_SOS.2.1 The TSF shall provide a mechanism to generate **authentication tokens** that satisfy [ [Table 24] Authentication Token Definition ].

FIA_SOS.2.2 The TSF shall enforce the use of **authentication tokens** generated by the TSF for [ general user integrated authentication ].

**[Table 24] Authentication Token Definition**

| Category | Description |
|---|---|
| **Authentication Token ID Length** | 36 (in bytes)<br>Web browser cookie storage, SSO server memory storage KEY value |
| **Authentication token structure** | JWT (Reference Standard: RFC7519) |
| **Header** | { alg: 'PS256', typ: 'JWT' } |

| | |
|---|---|
| **Authentication token configuration fields** | json Format:<br>Base64URL( {header} ).ARIA_ENC_Base64URL( {serverID, OTP(random), userID, expiration time(timestamp) } )<br>(ARIA_ENC_Base64URL: Base64URL format ARIA/CBC block cipher) |
| **Encryption Algorithm** | ARIA/CBC, 128bit |
| **Digital Signature Algorithm** | RSA-PSS (SHA-256) |
| **Server ID Length** | 6 (in bytes) |
| **User ID Length** | Minimum length: 3 characters or more, Maximum length: 50 characters or less |
| **OTP (Random Number) Length** | 12 (in bytes) |
| **Expiration time (timestamp) length** | 29 (in bytes) |
| **Signature Target Data Length** | 249 (in bytes) = Authentication Token Configuration Fields |
| **SSO server digital signature value length** | 384 (in bytes) |
| **Total Length of Authentication Token** | 762 (in bytes) = Data to be signed + '.' + SSO server electronic signature value |

#### 6.1.3.6. FIA_SOS.3 Destruction of secrets (Extended)

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

FIA_SOS.3.1 TSF shall destroy the **authentication token** according to the specified **authentication token** destruction method [Overwrite three times with the value '0'] that conforms to the following [None].

#### 6.1.3.7. FIA_UAU.2 Timing of authentication

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

FIA_UAU.2.1 TSF must successfully authenticate the user before allowing any other actions mediated by TSF on behalf of the user.

#### 6.1.3.8. FIA_UAU.4(1) Single-use Authentication Mechanisms (Administrator)

Hierarchical to : No other components.

Dependencies : No dependencies.

FIA_UAU.4.1 TSF shall prevent reuse of authentication data associated with [Administrator's CSRF token,[Table 25] Self-encoding payload ].

### 6.1.3.9. FIA_UAU.4(2) Single-use Authentication Mechanisms (General User)

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

FIA_UAU.4.1 TSF shall prevent reuse of authentication data related to [ General User CSRF Token, [Table 25] Self-encoding payload ].

**[Table 25] Self-encoding payload**

| **Self-encoding payload** | |
|---|---|
| Structure | (secret + hash(encode payload) + nonce) .(encode payload) |
| Encrypted with SSO server public key | secret + hash(encode payload) + nonce |
| SHA256 hash value of self-encoded payload | hash(encode payload) |
| secret | secret value |
| nonce | Random number |
| Self-encoded payload value | encode payload |
| payload | request data |

### 6.1.3.10. FIA_UAU.7 Protected authentication feedback

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

FIA_UAU.7.1 TSF shall only provide the user with [the password being entered displayed as "▪" instead of the input characters] during the authentication process.

### 6.1.3.11. FIA_UID.2 Timing of identification

Hierarchical to : FIA_UID.1 Identification
Dependencies : No dependencies.

FIA_UID.2.1 TSF must successfully identify each user before permitting any other actions mediated by TSF on behalf of the user.

### 6.1.4. Security Management (FMT)

#### 6.1.4.1. FMT_MOF.1 Security Function Management

Hierarchical to : No other components.

Dependencies : FMT_SMF.1 Specification of Management Functions.

FMT_SMR.1 Security Role.

FMT_MOF.1.1 TSF shall restrict the ability to determine ***management*** actions for the functions of the [Security Management Function] [ [Table 26] Security Management Functions ] to [Authorized Administrators].

**[Table 26] Security Management Functions**

| Subcategory | Security Management |
|---|---|
| Identification and authentication | User registration, deletion, modification, and authorization |
| Security management | IP registration, modification, and deletion for management terminals |
| | SSO Agent Inquiry - Status, Version, Applied Security Policy |
| | SSO Agent Security Policy Management - Policy Configuration |
| Self-protection | Performing integrity checks on TOE settings and the TOE itself upon administrator request |
| Update Protection | TOE Version Information Inquiry |
| Audit Log | Audit log inquiry |

### 6.1.4.2. FMT_MTD.1 Management of TSF data

Hierarchical to : No other components.

Dependencies : FMT_SMF.1 Specification of Management Functions.
FMT_SMR.1 Security Role.

FMT_MTD.1.1 TSF shall restrict the ability to ***manage*** the [ [Table 27] TSF Data List ] to [Authorized Administrators].

**[Table 27] TSF Data List**

| Related SFR | TSF Data Management Actions |
|---|---|
| FIA_UAU.1<br>FIA_UID.1 | Authorization assignment to user accounts (ID) |
| FIA_UAU.1<br>FIA_UID.1 | Addition and deletion of user IDs |
| FMT_MTD.1<br>FMT_PWD.1 | Addition, deletion, and modification of user passwords |
| FTA_TSE.1 | Registration, modification, deletion of management terminal IP addresses |
| FMT_MTD.1 | Agent inquiry - Mandatory inquiry information: agent version, applied security policy, agent operational status (enabled/disabled), agent integrity verification result (success/failure) |
| FMT_MTD.1 | Agent security policy management |
| FMT_MTD.1 | Configuration of authentication information for access by external IT entities |
| FMT_MTD.1 | Inquiry of identification information of the TOE and TOE components (e.g., server, agent) |
| FAU_SAR.1 | Inquiry of audit records |
| FPT_TST.1 | Execution of TOE configuration value verification and TOE self-integrity check upon administrator request |

### 6.1.4.3. 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 Role.

FMT_PWD.1.1 The TSF shall restrict the ability to manage the passwords defined in [ [Table 28] Password Function List ] to [ authorized administrators ] as follows:
1. Compliance requirements defined in [ [Table 29] Password Security Criteria

Type (1) ]
2. Prohibited items defined in [ [Table 29] Password Security Criteria Type (1) ]

FMT_PWD.1.2 The TSF shall restrict the ability to manage the IDs defined in [ [Table 30] ID Function List ] to [ authorized administrators ] as follows:
1. [ [Table 31] ID Composition Rules and Length ]
2. [ [Table 31] ID Security Requirements ]

FMT_PWD.1.3 The TSF shall provide the capability *for the authorized administrator to change the password upon first login*.

**[Table 28] Password Function List**

| No. | Function List | Remarks |
|---|---|---|
| 1 | Mandatory password change upon first login of administrator | |
| 2 | Password change when modifying administrator information | |
| 3 | General user password creation/change | |

**[Table 29] Password Security Criteria Type (1)**

| Category | Description |
|---|---|
| Compliance Requirements | Minimum length: 9 characters or more |
| | Maximum length: 255 characters or fewer |
| | Must include at least one digit, one uppercase letter (A–Z), one lowercase letter (a–z), and one special character |
| Prohibited Items | Prohibited from setting the password identical to the user account (ID) |
| | Prohibited from consecutive repetition of identical characters or numbers |
| | Prohibited from sequential input of adjacent keyboard characters or numbers |
| | Prohibited from reusing the previously used password |

**[Table 30] ID Function List**

| No. | Function | Remarks |
|---|---|---|
| 1 | General user ID creation | |

**[Table 31] ID Composition and Security Requirements**

| Category | Description |
|---|---|
| **ID Composition Rules** | ID Composition Rules Uppercase and lowercase letters (A-Z, a-z), digits (0-9) |
| | Prohibition of spaces and control characters (e.g., tab, newline) |

| | |
|---|---|
| | Restriction on consecutive identical characters (e.g., prevention of aaa, 1111) |
| | Restriction on common words to prevent dictionary attacks<br>: root, admin, user, test, manager |
| **ID Length Criteria** | Minimum length: 3 characters |
| | Maximum length: 50 characters |
| **Security Requirements** | Uniqueness: prevention of duplicate IDs within the same system |
| | Prohibition of ID identical to password: if the ID and password are identical, it is not permitted |

### 6.1.4.4. FMT_SMF.1 Specification of Management Functions

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

FMT_SMF.1.1 The TSF shall be able to perform the following management functions : [
a) Management of TSF functions : management functions specified in FMT_MOF.1
b) Management of TSF data : management functions specified in FMT_MTD.1
c) ID and password management : management functions specified in FMT_PWD.1
]

### 6.1.4.5. FMT_SMR.1 Security Role

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

FMT_SMR.1.1 The TSF shall maintain the roles defined in [ [Table 32] User classification and roles].

**[Table 32] User classification and roles**

| User Classification | Role | Description |
|---|---|---|
| Authorized Administrator | Super Administrator | Authorized administrator with all privileges (Read/Write) |
| Regular User | User | General user who can change their own password |

FMT_SMR.1.2 TSF shall be able to associate users and their **roles defined in FMT_SMR.1.1.**

### 6.1.5. Protection of the TSF(FPT)

#### 6.1.5.1. FPT_FLS.1 Maintenance of a secure state upon failure

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

FPT_FLS.1.1 The TSF shall maintain a secure state when the following type of failure occurs: [noise source health test failure].

#### 6.1.5.2. 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 the TSF data from *disclosure and modification* when it is transmitted between separate parts of the TOE.

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

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

FPT_PST.1.1 The TSF shall protect the following items stored in repositories controlled by the TSF [
a) passwords used for identification and authentication of administrators and general users
b) data encryption keys (DEK)
c) DB account identifiers (ID), passwords, and connection information used to access the DBMS
d) mail account identifiers (ID), passwords, and connection information used to access the mail server
e) RSA private keys used to authenticate the SSO server and SSO agent
f) TOE configuration values (security policies, configuration parameters, etc.)
g) integrity verification keys
h) authentication token encryption keys
i) session keys for transmission data encryption
] from unauthorized *disclosure*.

### 6.1.5.4. FPT_TST.1 TSF testing

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

FPT_TST.1.1 The TSF shall perform the following self-tests [ [Table 56] List of self-tests performed by the TSF ] *at start-up and periodically during normal operation* in order to demonstrate the correct operation of the *TSF*.

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 the *TSF*.

### 6.1.6. TOE access(FTA)

#### 6.1.6.1. FTA_MCS.2 Per user attribute limitation on multiple concurrent sessions

Hierarchical to : FTA_MCS.1 Basic Concurrent Session Limit
Dependencies : FIA_UID.1 Identification

FTA_MCS.2.1 TSF must limit the maximum number of concurrent sessions belonging to the same user according to the rule: [The maximum number of concurrent sessions for users with the same privileges and for the same user is limited to '1'. [If a user logs in from another terminal using the same account after logging in, new connections are blocked.]]

FTA_MCS.2.2 TSF must enforce a default session limit of [1] per user.

#### 6.1.6.2. FTA_SSL.3 TSF-initiated termination

Hierarchical to : No other components.
Dependencies : FMT_SMR.1 Security Role.

FTA_SSL.3.1 TSF shall terminate interactive sessions after [user inactivity period (fixed value: 10 minutes)].

#### 6.1.6.3. FTA_TSE.1 TOE session establishment

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

FTA_TSE.1.1 TSF must be able to reject **administrator management access session** settings based on [Connection IP, *None*].

### 6.1.7. Trusted path/Channels (FTP)

#### 6.1.7.1. FTP_ITC.1 Inter-TSF trusted channel

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

FTP_ITC.1.1 FTP_ITC.1.1 TSF shall provide an FTP_ITC communication channel that is logically

distinct from other communication channels between itself and other trusted IT products, provides assured identification of the terminal, and protects channel data from alteration or exposure.

FTP_ITC.1.2 The TSF shall permit *trusted IT* products to initiate communication through the secure channel.

FTP_ITC.1.3 The TSF must initiate communication via a secure channel for [sending notification emails when security alerts occur].

## 6.2. Security assurance requirements

Assurance requirements of this Protection Profile are comprised of assurance components in CC part 3, and the evaluation assurance level is EAL1+(ATE_FUN.1). The following table summarizes assurance components.

**[Table 33] Security assurance requirements**

| Security assurance class | Security assurance component | |
|---|---|---|
| Security Target evaluation | ASE_INT.1 | ST introduction |
| | ASE_CCL.1 | Conformance claims |
| | ASE_SPD.1 | Security problem definition |
| | 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 |

### 6.2.1. Security Target evaluation

#### 6.2.1.1. 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 In the case of a multi-assurance ST, the TOE overview shall describe the TSF composition with respect to the sub-TSFs defined in the PP-composition to which the ST claims conformance.

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

ASE_INT.1.9C 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.

### 6.2.1.2. 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 identify the CC publication to which the ST and the TOE claim conformance.

ASE_CCL.1.2C The 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 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 conformance claim shall be consistent with the extended components definition.

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

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

ASE_CCL.1.7C The conformance claim shall describe the conformance of the ST to a PP as PP-conformant.

ASE_CCL.1.8C The conformance claim rationale shall demonstrate that the TOE type of the ST is consistent with the TOE type of the PP-composition or PP to which the ST claims conformance.

ASE_CCL.1.9C The conformance claim rationale shall demonstrate that the statement of the security problem definition of the ST is consistent with the statement of the security problem definition of the PP-composition, PP, and functional packages to which the ST claims conformance.

ASE_CCL.1.10C The conformance claim rationale shall demonstrate that the statement of the security objectives of the ST is consistent with the statement of the security objectives of the PP-composition, PP, and functional packages to which the ST claims conformance.

ASE_CCL.1.11C The conformance claim rationale shall demonstrate that the statement of the security requirements of the ST is consistent with the statement of the security requirements of the PP-composition, PP, and functional packages to which the ST claims conformance.

ASE_CCL.1.12C The conformance claim to a PP or PP-composition shall be exact conformance, strict conformance, demonstrable conformance, or a list of conformance types.

ASE_CCL.1.13C If the conformance claim identifies a set of evaluation methods and evaluation activities derived from the CEM work units to be used for the TOE evaluation, this set shall include all those contained in the packages, PPs, or PP-modules of the PP-composition to which the ST claims conformance, and no others shall be permitted.

Evaluator action elements

ASE_CCL.1.1E The evaluator shall confirm that the information provided meets all evidence requirements.

### 6.2.1.3. ASE_SPD.1 Security problem definition

Dependencies : No dependencies.

Developer action elements

ASE_SPD.1.1D The developer shall provide a security problem definition.

Content and presentation elements

ASE_SPD.1.1C The security problem definition shall describe the threats.

ASE_SPD.1.2C All threats shall be described in terms of the threat agent, the asset, and the adverse action.

ASE_SPD.1.3C The security problem definition shall describe the organizational security policies.

ASE_SPD.1.4C The security problem definition shall describe the assumptions regarding the operational environment of the TOE.

Evaluator action elements

ASE_SPD.1.1E The evaluator shall confirm that the information provided meets all evidence requirements.

### 6.2.1.4. ASE_OBJ.1 Security objectives for the operational environment

Dependencies : ASE_SPD.1 Security problem definition

Developer action elements

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

ASE_OBJ.1.2D The developer shall provide a rationale for the security objectives for the operational environment.

Content and presentation elements

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

ASE_OBJ.1.2C The security objectives rationale shall trace each security objective for the operational environment to the threats countered by the security objectives, the organizational security policies enforced by the security objectives, and the assumptions supported by the security objectives.

ASE_OBJ.1.3C The security objectives rationale shall demonstrate that the security objectives for the operational environment support all assumptions.

Evaluator action elements

ASE_OBJ.1.1E The evaluator shall confirm that the information provided meets all evidence requirements.

### 6.2.1.5. 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 to each element can be demonstrated.

Evaluator action elements

ASE_ECD.1.1E The evaluator shall confirm that the information provided meets all evidence requirements.

ASE_ECD.1.2E The evaluator shall confirm that the extended components cannot be clearly expressed using existing CC components.

### 6.2.1.6. ASE_REQ.1 Stated security requirements

Dependencies : ASE_ECD.1 Extended components definition
ASE_SPD.1 Security problem definition
ASE_OBJ.1 Security objectives for the operational environment

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 In the case of a single-assurance ST, the statement of security requirements shall define a global set of SARs applicable to the entire TOE. The set of SARs shall be consistent with the PP or PP-composition to which the ST claims conformance.

ASE_REQ.1.3C In the case of a multi-assurance ST, the statement of security requirements shall define a global set of SARs applicable to the entire TOE and sets of SARs applicable to the sub-TSFs. The sets of SARs shall be consistent with the multi-assurance PP-composition to which the ST claims conformance.

ASE_REQ.1.4C All subjects, objects, operations, security attributes, external entities, and other terms used in the SFRs and SARs shall be defined.

ASE_REQ.1.5C The statement of security requirements shall identify all operations on the security requirements.

ASE_REQ.1.6C All operations shall be correctly performed.

ASE_REQ.1.7C All dependencies of the security requirements shall be satisfied; otherwise, justification shall be provided in the security requirements rationale.

ASE_REQ.1.8C The security requirements rationale shall trace each SFR to the threats countered by the SFR and the organizational security policies enforced by the SFR.
The security requirements rationale shall demonstrate that the SFRs, together with the security objectives for the operational environment, counter all threats to the TOE.

ASE_REQ.1.9C The security requirements rationale shall demonstrate that the SFRs, together with the security objectives for the operational environment, enforce all organizational security policies for the TOE.

ASE_REQ.1.10C The security requirements rationale shall explain the reasons for selecting the SARs.

ASE_REQ.1.11C The statement of security requirements shall be internally consistent.

ASE_REQ.1.12C If the ST defines a set of SARs that extends the set of SARs of a PP or PP-composition to which the ST claims conformance, the security requirements rationale shall include an assurance rationale that justifies the consistency of the extension and provides a rationale for the treatment of the evaluation methods and evaluation activities identified in the conformance method that are affected by the extension of the set of SARs.

Evaluator action elements

ASE_REQ.1.1E The evaluator shall confirm that the information provided meets all evidence requirements.

### 6.2.1.7. 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 evidence requirements.

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

### 6.2.2. Development

#### 6.2.2.1. 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 a rationale for the classification of interfaces as SFR-non-interfering.

ADV_FSP.1.4C The traceability shall demonstrate that the SFRs are traced to the TSFIs within the functional specification.

Evaluator action elements

ADV_FSP.1.1E The evaluator shall confirm that the information provided meets all evidence requirements.

ADV_FSP.1.2E The evaluator shall determine that the functional specification accurately and completely instantiates the SFRs.

### 6.2.3. Guidance documents

#### 6.2.3.1. 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 functions and privileges accessible to the user that are controlled within 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 interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3C The operational user guidance shall describe, for each user role, the functions and interfaces available. In particular, it shall clearly indicate the secure values for all security parameters under the control of the user.

AGD_OPE.1.4C The operational user guidance shall clearly present, for each user role, each type of security-relevant event associated with user-accessible functions that shall be performed, including changes to 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 a failure or following an operational error), their consequences, and any related matters necessary to maintain secure operation.

AGD_OPE.1.6C The operational user guidance shall describe, for each user role, the security controls that shall be observed in order to satisfy 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 evidence requirements.

### 6.2.3.2. AGD_PRE.1 Preparative procedures

Dependencies : No dependencies.

Developer action elements

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

Content and presentation elements

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

AGD_PRE.1.2C The preparative procedures shall describe all steps necessary for the secure

installation of the TOE and the secure preparation of the operational environment, consistent with the security objectives for the operational environment described in the ST.

Evaluator action elements

AGD_PRE.1.1E The evaluator shall confirm that the information provided meets all evidence requirements.

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

### 6.2.4. Life-cycle support

#### 6.2.4.1. ALC_CMC.1 Labelling of the TOE

Dependencies : ALC_CMS.1 TOE CM coverage

Developer action elements

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

Content and presentation elements

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

Evaluator action elements

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

#### 6.2.4.2. 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 TOE 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 evidence

requirements.

### 6.2.5. Tests

#### 6.2.5.1. 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 the test documentation.

Content and presentation elements

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

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

ATE_FUN.1.3C The expected test results shall describe the results expected from the successful execution of the tests.

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

Evaluator action elements

ATE_FUN.1.1E The evaluator shall confirm that the information provided meets all evidence requirements.

#### 6.2.5.2. ATE_IND.1 Independent testing : conformance

Dependencies : ADV_FSP.1 Basic functional specification
AGD_OPE.1 Operational User Guidance
AGD_PRE.1 Preparative Procedures

Developer action elements

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

Content and presentation elements

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

Evaluator action elements

ATE_IND.1.1E The evaluator shall confirm that the information provided satisfies all evidence

requirements.

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

### 6.2.6. Vulnerability assessment

#### 6.2.6.1. 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 evidence requirements.

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

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

## 6.3. Rationale for Security Requirements

The rationale for the security requirements demonstrates that the stated Security Functional Requirements (SFRs) are suitable to satisfy the security objectives and, as a result, are appropriate to address the security problem definition. In addition, the dependencies of the EAL1 assurance package provided by the Common Criteria for Information Technology Security Evaluation are already satisfied.

### 6.3.1. Rationale for Security Functional Requirements

The rationale for the Security Functional Requirements demonstrates the following:

- Each threat and organizational security policy is addressed by at least one Security Functional Requirement.
- Each Security Functional Requirement is traceable to at least one threat or organizational security policy.

The table below shows the correspondence between the security objectives and the Security Functional Requirements.

**[Table 34] Mapping Between Security Objectives and Security Functional Requirements**

| Threats and/or Organizational / SFR | Threats | | | | | | | | | Organizational Security Policies | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | T.SESSION_HIJACK | T.RETRY_AUTH_ATTEMPT | T.IMPERSONATION | T.REPLAY | T.WEAK_PASSWORD | T.STORED_DATA_LEAKAGE | T.TRANSMISSION_DATA_DAMAGE | T.WEAK_CRYPTO_PROTOCOLS | T.TSF_COMPROMISE | P.AUDIT | P.SECURE_OPERATION | P.CRYPTO_STRENGTH |
| FAU_ARP.1 | | | | | | | | | X | | | |
| FAU_GEN.1 | | | | | | | | | | X | | |
| FAU_SAA.1 | | | | | | | | | X | | | |
| FAU_SAR.1 | | | | | | | | | | X | | |
| FAU_SAR.3 | | | | | | | | | | X | | |
| FAU_STG.1 | | | | | | | | | | X | | |
| FAU_STG.4 | | | | | | | | | | X | | |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FAU_STG.5 | | | | | | | | | | X | | |
| FCS_CKM.1 | | | | | | X | X | X | | | | X |
| FCS_CKM.2 | | | | | | X | X | X | | | | X |
| FCS_CKM.5 | | | | | | X | X | X | | | | X |
| FCS_CKM.6 | | | | | | X | X | X | | | | X |
| FCS_COP.1 | | | | | | X | X | X | | | | X |
| FCS_RBG.1 | | | | | | X | X | X | | | | X |
| FCS_RBG.3 | | | | | | X | X | X | | | | X |
| FIA_AFL.1(1) | | X | X | | | | | | X | | | |
| FIA_AFL.1(2) | | X | X | | | | | | X | | | |
| FIA_IMA.1 | | | X | | | | | | | | | |
| FIA_SOS.1 | | | | | X | | | | | | | |
| FIA_SOS.2 | | | X | X | | | | | | | | |
| FIA_SOS.3 | X | | X | | | | | | | | | |
| FIA_UAU.2 | | | X | | | | | | X | | | |
| FIA_UAU.4(1) | | | X | X | | | | | X | | | |
| FIA_UAU.4(2) | | | X | X | | | | | X | | | |
| FIA_UAU.7 | | | X | | X | | | | X | | | |
| FIA_UID.2 | | | X | | | | | | X | | | |
| FMT_MOF.1 | | | | | | | | | X | | X | |
| FMT_MTD.1 | | | | | | | | | X | | X | |
| FMT_PWD.1 | | | | | X | | | | X | | X | |
| FMT_SMF.1 | | | | | | | | | X | | X | |
| FMT_SMR.1 | | | | | | | | | X | | X | |
| FPT_FLS.1 | | | | | | X | X | X | | | | X |
| FPT_ITT.1 | | | | | | | X | | | | | |
| FPT_PST.1 | | | | | | X | | | | | | |
| FPT_TST.1 | | | | | | X | X | X | X | | | X |
| FTA_MCS.2 | X | | | | | | | | | | | |
| FTA_SSL.3 | X | | | | | | | | | | | |
| FTA_TSE.1 | X | | | | | | | | | | | |
| FTP_ITC.1 | | | | | | | X | | | | | |

**T.SESSION_HIJACK** FIA_SOS.3, FTA_MCS.2, FTA_SSL.3, FTA_TSE.1

FIA_SOS.3 ensures that when a user session is terminated, the session storing the authentication token is invalidated and the related data are initialized to '0', thereby ensuring destruction of the authentication token; therefore, it addresses T.SESSION_HIJACK.

FTA_MCS.2 restricts duplicate access to the TOE using the same user account or the same privileges;

therefore, it addresses T.SESSION_HIJACK.

FTA_SSL.3 ensures session locking or session termination of interactive sessions after a period of inactivity by an authorized user; therefore, it addresses T.SESSION_HIJACK.

FTA_TSE.1 ensures that the TOE determines whether to establish an authorized user session based on IP address or other attributes; therefore, it addresses T.SESSION_HIJACK.

---

| **T.RETRY_AUTH_ATTEMPT** | FIA_AFL.1(1), FIA_AFL.1(2) |
|---|---|

FIA_AFL.1(1) and FIA_AFL.1(2) define the maximum number of failed authentication attempts for authorized administrators and general users and ensure that response actions are taken when the defined threshold is reached; therefore, they address T.RETRY_AUTH_ATTEMPT.

---

| **T.IMPERSONATION** | FIA_AFL.1(1), FIA_AFL.1(2), FIA_IMA.1, FIA_SOS.2, FIA_SOS.3, FIA_UAU.2, FIA_UAU.4(1), FIA_UAU.4(2), FIA_UAU.7, FIA_UID.2 |
|---|---|

FIA FIA_AFL.1(1) and FIA_AFL.1(2) define the maximum number of failed authentication attempts for authorized administrators and general users and ensure that response actions are taken when the defined threshold is reached; therefore, they address T.IMPERSONATION.

FIA_IMA.1 ensures that mutual authentication is performed between TOE components; therefore, it addresses T.IMPERSONATION.

FIA_SOS.2, FIA_SOS.3, FIA_UAU.2, FIA_UAU.4(1), and FIA_UAU.4(2) ensure successful authentication of administrators and general users attempting to access the TOE; therefore, they address T.IMPERSONATION.

FIA_UAU.7 ensures that only masked values are displayed or that values are not displayed during the authentication process, and that no feedback is provided regarding the reason for authentication failure; therefore, it addresses T.IMPERSONATION.

FIA_UID.2 ensures successful identification of administrators and general users attempting to access the TOE; therefore, it addresses T.IMPERSONATION.

---

| **T.REPLAY** | FIA_SOS.2, FIA_UAU.4(1), FIA_UAU.4(2) |
|---|---|

FIA_SOS.2 ensures prevention of authentication token reuse when generating authentication tokens; therefore, it addresses T.REPLAY.

---

FIA_UAU.4(1) and FIA_UAU.4(2) ensure the capability to prevent reuse of authentication data; therefore, they address T.REPLAY.

---

| | |
|---|---|
| **T.WEAK_PASSWORD** | FIA_SOS.1, FIA_UAU.7, FMT_PWD.1 |

FIA_SOS.1 verifies that password complexity rules are satisfied; therefore, it addresses T.WEAK_PASSWORD.

FIA_UAU.7 ensures that only masked values are displayed or that values are not displayed during authentication; therefore, it addresses T.WEAK_PASSWORD.

FMT_PWD.1 ensures the capability to enforce mandatory password change for the default password upon the first login of an authorized administrator; therefore, it addresses T.WEAK_PASSWORD.

---

| | |
|---|---|
| **T.STORED_DATA_LEAKAGE** | FCS_CKM.1, FCS_CKM.2, FCS_CKM.5, FCS_CKM.6, FCS_COP.1, FCS_RBG.1, FCS_RBG.3, FPT_FLS.1, FPT_PST.1, FPT_TST.1 |

FCS_CKM.1, FCS_CKM.2, FCS_CKM.5, FCS_RBG.1, FCS_RBG.3, FPT_FLS.1, and FPT_TST.1 ensure that cryptographic keys are generated and distributed in accordance with secure cryptographic algorithms and key lengths when encrypting stored data; therefore, they address T.STORED_DATA_LEAKAGE.

FCS_CKM.6 ensures that cryptographic keys and related information are destroyed in accordance with the specified destruction method upon completion of stored data encryption; therefore, it addresses T.STORED_DATA_LEAKAGE.

FCS_COP.1 ensures that cryptographic operations are performed using the specified secure algorithms and key lengths when encrypting stored data; therefore, it addresses T.STORED_DATA_LEAKAGE.

FPT_PST.1 ensures that TSF data stored using cryptographic mechanisms and access control are protected against disclosure threats; therefore, it addresses T.STORED_DATA_LEAKAGE.

---

| | |
|---|---|
| **T.TRANSMISSION_DATA_DAMAGE** | FCS_CKM.1, FCS_CKM.2, FCS_CKM.5, FCS_CKM.6, FCS_COP.1, FCS_RBG.1, FCS_RBG.3, FPT_FLS.1, FPT_ITT.1, FTP_ITC.1 |

cryptographic keys are generated and distributed in accordance with secure cryptographic algorithms and key lengths during cryptographic communication; therefore, they address T.TRANSMISSION_DATA_DAMAGE.

---

FCS_CKM.6 ensures that cryptographic keys and related information are destroyed in accordance with the specified destruction method upon termination of cryptographic communication; therefore, it addresses T.TRANSMISSION_DATA_DAMAGE.

FCS_COP.1 ensures that cryptographic operations are performed using the specified secure algorithms and key lengths during cryptographic communication; therefore, it addresses T.TRANSMISSION_DATA_DAMAGE.

FPT_ITT.1 ensures the confidentiality and integrity of data transmitted between TOE components; therefore, it addresses T.TRANSMISSION_DATA_DAMAGE.

FTP_ITC.1 ensures the confidentiality and integrity of data transmitted between the TOE and external IT entities; therefore, it addresses T.TRANSMISSION_DATA_DAMAGE.

---

| | |
|---|---|
| **T.WEAK_CRYPTO_PROTOCOLS** | FCS_CKM.1, FCS_CKM.2, FCS_CKM.5, FCS_CKM.6, FCS_COP.1, FCS_RBG.1, FCS_RBG.3, FPT_FLS.1, FPT_TST.1 |

FCS_CKM.1, FCS_CKM.2, FCS_CKM.5, FCS_RBG.1, FCS_RBG.3, FPT_FLS.1, and FPT_TST.1 ensure that cryptographic keys are generated and distributed in accordance with the required cryptographic algorithms and key lengths for standard cryptographic algorithms with a security strength of at least 112 bits when encrypting transmitted data; therefore, they address T.WEAK_CRYPTO_PROTOCOLS.

FCS_CKM.6 ensures that cryptographic keys and related information are destroyed in accordance with the specified destruction method; therefore, it addresses T.WEAK_CRYPTO_PROTOCOLS.

FCS_COP.1 ensures that cryptographic operations are performed using standard algorithms with a security strength of at least 112 bits and appropriate key lengths when encrypting transmitted data; therefore, it addresses T.WEAK_CRYPTO_PROTOCOLS.

---

| | |
|---|---|
| **T.TSF_COMPROMISE** | FAU_ARP.1, FAU_SAA.1, FIA_AFL.1(1), FIA_AFL.1(2), FIA_UAU.2, FIA_UAU.4(1), FIA_UAU.4(2), FIA_UAU.7, FIA_UID.2, FMT_MOF.1, FMT_MTD.1, FMT_PWD.1, FMT_SMF.1, FMT_SMR.1, FPT_TST.1 |

FAU_ARP.1 ensures that response actions are taken when security violations, such as compromise of TOE integrity, are detected; therefore, it addresses T.TSF_COMPROMISE.

FAU_SAA.1 ensures the capability to analyze audited events to detect indications of security violations, including compromise of TOE integrity; therefore, it addresses T.TSF_COMPROMISE.

---

FIA_AFL.1(1), FIA_AFL.1(2), FIA_UAU.2, FIA_UAU.4(1), FIA_UAU.4(2), FIA_UAU.7, and FIA_UID.2 ensure that access to the TOE is permitted only after successful user identification and authentication, thereby preventing unauthorized bypass access by threat agents; therefore, they address T.TSF_COMPROMISE.

FMT_MOF.1, FMT_MTD.1, FMT_PWD.1, FMT_SMF.1, and FMT_SMR.1 categorize authorized user roles as administrators and general users for access to and configuration of management functions, and provide security policies and security functions according to roles, thereby preventing unauthorized access by threat agents; therefore, they address T.TSF_COMPROMISE.

FPT_TST.1 ensures self-testing of the TSF for correct operation of the TOE and provides the capability for authorized administrators to verify the integrity of TSF data and the TSF itself; therefore, it addresses T.TSF_COMPROMISE.

---

| | |
|---|---|
| **P.AUDIT** | FAU_GEN.1, FAU_SAR.1, FAU_SAR.3, FAU_STG.1, FAU_STG.4, FAU_STG.5 |

FAU_GEN.1 ensures that audit records are generated for auditable events such as the start-up and shutdown of the audit function, and the success or failure of identification and authentication of administrators; therefore, it satisfies P.AUDIT.

FAU_SAR.1 provides authorized administrators with the capability to review audit records and ensures that audit records are presented in a manner suitable for interpretation by administrators; therefore, it satisfies P.AUDIT.

FAU_SAR.3 provides the capability to perform selective audit review of audit data based on logical relationship criteria; therefore, it satisfies P.AUDIT.

FAU_STG.1, in the case of the TOE Server, provides the capability to store audit data in local storage or to transmit it in real time to an external IT entity over a secure channel for storage; therefore, it satisfies P.AUDIT.

FAU_STG.4 ensures that appropriate response actions are taken when the audit trail of the TOE Server exceeds the storage capacity threshold; therefore, it satisfies P.AUDIT.

FAU_STG.5 ensures that appropriate response actions are taken when the audit trail of the TOE Server becomes full; therefore, it satisfies P.AUDIT.

---

| | |
|---|---|
| **P.SECURE_OPERATION** | FMT_MOF.1, FMT_MTD.1, FMT_PWD.1, FMT_SMF.1, FMT_SMR.1 |

FMT_MOF.1 FMT_MOF.1 ensures that only authorized users are able to manage security functions; therefore, it satisfies P.SECURE_OPERATION.

FMT_MTD.1 ensures that only authorized users are able to manage TSF data; therefore, it satisfies P.SECURE_OPERATION.

FMT_PWD.1 ensures that only authorized administrators are able to manage ID and password composition rules and length requirements, and provides functionality such as mandatory password change upon first login by an authorized user; therefore, it satisfies P.SECURE_OPERATION.

FMT_SMF.1 requires specification of the management functions for security functions, security attributes, and TSF data that the TSF must perform; therefore, it satisfies P.SECURE_OPERATION.

FMT_SMR.1 ensures that authorized roles related to security management are specified; therefore, it satisfies P.SECURE_OPERATION.

| | |
|---|---|
| **P.CRYPTO_STRENGTH** | FCS_CKM.1, FCS_CKM.2, FCS_CKM.5, FCS_CKM.6, FCS_COP.1, FCS_RBG.1, FCS_RBG.3, FPT_FLS.1, FPT_TST.1 |

FCS_CKM.1, FCS_CKM.2, FCS_CKM.5, FCS_CKM.6, FCS_COP.1, FCS_RBG.1, FCS_RBG.3, FPT_FLS.1, FPT_FLS.1, and FPT_TST.1 ensure that cryptographic keys required for standard cryptographic algorithms with a security strength of at least 112 bits are securely generated and distributed when encrypting data; therefore, they satisfy P.CRYPTO_STRENGTH.

FCS_COP.1 ensures that cryptographic operations are performed using standard algorithms with a security strength of at least 112 bits and appropriate key lengths when encrypting data; therefore, it satisfies P.CRYPTO_STRENGTH.

### 6.3.2. Security assurance requirements rationale

The Evaluation Assurance Level selected for this Security Target is EAL1+(ATE_FUN.1).

EAL1 may be applied where some confidence in correct operation is required but the security threats are not considered serious. If the TOE has been developed in accordance with commonly applied development methodologies, EAL1 does not require additional developer effort to prepare evaluation evidence. In other words, it does not require significant additional cost or time investment for evaluation preparation.

EAL1 provides a basic level of assurance by analyzing the Security Functional Requirements included in a limited Security Target through functional and interface specifications and guidance documentation in order to understand security behavior.

This analysis is supported by independent testing of the TSF and searches for publicly known vulnerabilities (functional testing and penetration testing).

Although EAL1 does not require evidence of developer testing based on the functional specification, this Security Target additionally includes ATE_FUN.1 so that the developer performs testing to verify whether the TSF has been correctly implemented and whether defects occur, and documents the results.

## 6.4. Rationale for dependencies

### 6.4.1. Dependencies of the security functional requirements

The Security Functional Requirements used in this Security Target show dependencies as indicated in the table below.

**[Table 35] Dependencies of Security Functional Requirements**

| No. | Security Functional Requirement | Dependency | Reference No. |
|---|---|---|---|
| 1 | FAU_ARP.1 | FAU_SAA.1 | 3 |
| 2 | FAU_GEN.1 | FPT_STM.1 | Rationale (1) |
| 3 | FAU_SAA.1 | FAU_GEN.1 | 2 |
| 4 | FAU_SAR.1 | FAU_GEN.1 | 2 |
| 5 | FAU_SAR.3 | FAU_SAR.1 | 4 |
| 6 | FAU_STG.1 | FAU_GEN.1 | 2 |
| | | FTP_ITC.1 | Rationale (2) |
| 7 | FAU_STG.4 | FAU_STG.2 | Rationale (3) |
| 8 | FAU_STG.5 | FAU_STG.2 | Rationale (3) |
| 9 | FCS_CKM.1 | [FCS_CKM.2 or FCS_CKM.5 or FCS_COP.1] | 10, 11 |
| | | [FCS_RBG.1 or FCS_RNG.1] | 14 |
| | | FCS_CKM.6 | 12 |
| 10 | FCS_CKM.2 | [FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5] | 9 |
| 11 | FCS_CKM.5 | [FCS_CKM.2 or FCS_COP.1] | 10, 13 |
| | | FCS_CKM.6 | 12 |
| 12 | FCS_CKM.6 | [FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or | 9 |

| | | | |
|---|---|---|---|
| | | FCS_CKM.5] | |
| 13 | FCS_COP.1 | [FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5] | 9 |
| | | FCS_CKM.6 | 12 |
| 14 | FCS_RBG.1 | [FCS_RBG.2 or FCS_RBG.3] | 15 |
| | | FPT_FLS.1 | 32 |
| | | FPT_TST.1 | 35 |
| 15 | FCS_RBG.3 | FCS_RBG.1 | 14 |
| 16 | FIA_AFL.1(1) | FIA_UAU.1 | 22 |
| 17 | FIA_AFL.1(2) | FIA_UAU.1 | 22 |
| 18 | FIA_IMA.1 | - | - |
| 19 | FIA_SOS.1 | - | - |
| 20 | FIA_SOS.2 | - | - |
| 21 | FIA_SOS.3 | FIA_SOS.2 | 20 |
| 22 | FIA_UAU.2 | FIA_UID.1 | 26 |
| 23 | FIA_UAU.4(1) | - | - |
| 24 | FIA_UAU.4(2) | - | - |
| 25 | FIA_UAU.7 | FIA_UAU.1 | 22 |
| 26 | FIA_UID.2 | - | - |
| 27 | FMT_MOF.1 | FMT_SMF.1 | 30 |
| | | FMT_SMR.1 | 31 |
| 28 | FMT_MTD.1 | FMT_SMF.1 | 30 |
| | | FMT_SMR.1 | 31 |
| 29 | FMT_PWD.1 | FMT_SMF.1 | 30 |
| | | FMT_SMR.1 | 31 |
| 30 | FMT_SMF.1 | - | - |
| 31 | FMT_SMR.1 | FIA_UID.1 | 26 |
| 32 | FPT_FLS.1 | - | - |
| 33 | FPT_ITT.1 | - | - |
| 34 | FPT_PST.1 | - | - |
| 35 | FPT_TST.1 | - | - |
| 36 | FTA_MCS.2 | FIA_UID.1 | 26 |
| 37 | FTA_SSL.3 | FMT_SMR.1 | 31 |
| 38 | FTA_TSE.1 | - | - |
| 39 | FTP_ITC.1 | - | - |

Theoretical Rationale (1): FAU_GEN.1 has a dependency on FPT_STM.1. This dependency is satisfied by the security objective for the operational environment, OE.TrustedTimestamp, since security-related events are recorded using a reliable timestamp provided by the TOE operational environment.

Theoretical Rationale (2): FAU_STG.1 has a dependency on FTP_ITC.1. This dependency is satisfied by the operational environment security objective OE.SecureDBMS.

Theoretical Rationale (3): FAU_STG.4 and FAU_STG.5 have a dependency on FAU_STG.2. This dependency is satisfied by the operational environment security objective OE.SecureDBMS.

FIA_AFL.1(1), FIA_AFL.1(2), and FIA_UAU.7 have a dependency on FIA_UAU.1. This dependency is satisfied by FIA_UAU.2, which is hierarchically related to FIA_UAU.1.

FIA_UAU.2, FMT_SMR.1, and FTA_MCS.2 have a dependency on FIA_UID.1. This dependency is satisfied by FIA_UID.2, which is hierarchically related to FIA_UID.1.

### 6.4.2. Dependencies of the security assurance requirements

The dependencies of the EAL1 assurance package provided by the Common Criteria for Information Technology Security Evaluation are already satisfied; therefore, the theoretical rationale for these dependencies is omitted.
The additionally included assurance requirement ATE_FUN.1 has a dependency on ATE_COV.1. ATE_FUN.1 was included to ensure that the developer correctly performs testing for the test items and records the results in the test documentation. However, it was determined that ATE_COV.1, which demonstrates the correspondence between test items and TSFIs, is not mandatory; therefore, it was not included in this Security Target.

Although this Security Target conforms to the EAL1 assurance package, ASE_OBJ.1 includes ASE_SPD.1 as a dependency, which is not included in the EAL1 assurance package. However, this Direct Rationale Protection Profile includes the security problem definition, and ASE_OBJ.1 provides indirect assurance of the security problem definition by requiring examination of whether the security objectives for the TOE operational environment are traceable to the security problem definition. Therefore, it was determined that ASE_SPD.1, which relates to the description requirements of the security problem definition, is not mandatory and was not included in this Security Target.

ASE_REQ.1 also includes ASE_SPD.1 as a dependency that is not present in the EAL1 assurance package. However, this Direct Rationale Security Target includes the security problem definition, and ASE_REQ.1 provides indirect assurance of the security problem definition by requiring examination of whether the SFRs are traceable to the security problem definition. Therefore, it was determined that ASE_SPD.1 is not mandatory and was not included in this Security Target.

# 7. TOE summary specification

This section briefly and clearly describes how the security functions of the TOE are implemented.

## 7.1. Security Audit

The TOE's security audit function consists of the following security functionalities: security alarm, audit data generation, potential violation analysis, audit review, selectable audit review, protection of audit data storage, actions in case of possible audit data loss, and prevention of audit data loss.

### 7.1.1. Security Alarm

The TOE shall generate a security alarm when an administrator or general user authentication attempt fails consecutively five (5) times (fixed value), and shall send an email notification to the authorized administrator.

The SSO Server shall, upon start-up or upon invocation of the periodic self-test (12-hour interval), terminate the server process, store an audit record, and notify the authorized administrator via email if the self-test result indicates failure.

The SSO Agent shall, upon start-up or upon invocation of the periodic self-test (12-hour interval), transmit the audit record to the SSO Server and notify the authorized administrator via email if the self-test result indicates failure.

The TOE shall store an audit record and notify the authorized administrator via email when a manually invoked integrity test by an authorized administrator results in failure.

The TOE shall store an audit record and notify the authorized administrator via email when the audit evidence storage capacity exceeds the defined threshold (fixed value: 80%).

The TOE shall, when the audit evidence storage reaches a saturated state (fixed value: 90%), store a single audit record indicating the saturated condition, ignore subsequently auditable events while the storage remains saturated, and notify the authorized administrator via email.

Related SFR: FAU_ARP.1, FAU_SAA.1

### 7.1.2. Audit Data Generation

Each TOE component generates audit data for security-relevant events, and the generated audit

data is stored in the DBMS, which is an external IT entity provided by the operational environment. Among the TOE components, audit data generated by the SSO Agent is transmitted to the SSO Server, and the SSO Server stores it in the DBMS, while audit data generated by the SSO Server is stored directly in the DBMS. The TOE uses a trusted timestamp provided by the operational environment co-located with the SSO Server in order to maintain the chronological order of the generated audit data.

The audit data consists of the date and time of the event, the type of event, the identity of the subject (if available), the subject's IP address, the outcome of the event (success or failure), and additional audit records (if additional audit record content exists).

Audit data is generated for the auditable events defined in ' [Table 36] Auditable Events '

**[Table 36] Auditable Events**

| Category | Auditable Event | Additional Audit Record Content |
|---|---|---|
| Security Audit | Start-up and shutdown of SSO Server and SSO Agent | |
| | Actions taken and results when audit storage fails | |
| Cryptographic Support | Cryptographic key generation failure | |
| | Cryptographic operation failure (including cryptographic operation type) | |
| Identification and Authentication | User login, logout | |
| | User registration, modification, deletion | |
| | Actions taken upon reaching authentication attempt limit | |
| | All password changes | |
| | Success/Failure of mutual authentication between SSO Server and SSO Agent | |
| | SSO authentication token generation or verification result (success/failure) | |
| | SSO authentication token destruction and its result (success/failure) | |
| Security Management | Registration, modification, deletion of management terminal access IP | |
| | Execution of security management functions and all changes/deletions of security attribute values | Modified security attribute data |
| | Default account password change | |

| | | |
|---|---|---|
| | Change of agent registration status | |
| TSF Protection | SSO Server and SSO Agent self-test | Failed security function |
| | SSO Server and SSO Agent integrity verification | Component that failed integrity verification |
| TOE Access | User session termination | |
| | Action taken upon detection of duplicate login attempts for the same account | |
| | Rejection of new session based on concurrent session limit | |
| | Blocking of management terminal access IP | |

Related SFR: FAU_GEN.1, FAU_STG.1

### 7.1.3. Potential Violation Analysis

The TSF performs potential security violation detection based on the generated and collected security audit data (refer to 6.1.1 Security Audit (FAU)). When a potential security violation defined in the Security Target is detected, the TSF performs the response actions by means of a security alarm as specified in "[Table 37] Actions for Security Violations."

**[Table 37] Actions for Security Violations**

| Security Violation | Response Action |
|---|---|
| Administrator authentication attempts fail consecutively five (5) times (fixed value) | - Disable authentication function for a defined period (fixed value: 5 minutes)<br>- Send warning email to authorized administrator |
| General user authentication attempts fail consecutively five (5) times (fixed value) | - Disable authentication function for a defined period (fixed value: 5 minutes)<br>- Send warning email to authorized administrator |
| SSO Server start-up or periodic self-test failure | - Terminate server process<br>- Send warning email to authorized administrator |
| SSO Agent start-up or periodic self-test failure | - Send warning email to authorized administrator |
| Manual integrity test failure invoked by authorized administrator | - Send warning email to authorized administrator |
| Audit evidence storage exceeds threshold (fixed value: 80%) | - Send warning email to authorized administrator |

| | |
|---|---|
| Audit evidence storage reaches saturation (fixed value: 90%) | - Send warning email to authorized administrator<br>- Store a single saturation audit record and prevent further use of audit records<br>- Ignore subsequently auditable events |

Related SFR: FAU_SAA.1, FAU_ARP.1

### 7.1.4. Audit Review

When auditable events occur, the audit data is stored in the local DBMS. An authorized administrator accesses the Web GUI-based security management interface provided by the SSO Server and, after successful identification and authentication using administrator ID and password, selects the defined audit record inquiry and search menu through the "Audit > Activity Audit" menu to review the audit data.

Related SFR: FAU_SAR.1, FAU_STG.1

### 7.1.5. Selectable Audit Review

In the Web GUI-based audit record inquiry menu provided by the SSO Server, audit data can be sorted in descending order based on event date and time, and audit records can be queried by selecting event date/time and event type.

Related SFR: FAU_SAR.3

### 7.1.6. Actions in Case of Possible Audit Data Loss

Audit records generated by the TOE are stored in the storage provided by the operational environment (DBMS). Administrators with access privileges to the audit record DB may perform maintenance operations through the storage.

The total size of the storage space for audit evidence is based on the total size of the tablespace configured for the audit table when the TOE-specific DB account is created in the operational environment (DBMS).

The TOE periodically checks the audit record storage, and when the audit records stored in the DBMS exceed the defined threshold (fixed value: 80%), the TOE stores an audit record indicating threshold exceedance and sends a warning email to the authorized administrator.

Related SFR: FAU_STG.4

### 7.1.7. Prevention of Audit Data Loss

When the audit records stored in the DBMS reach saturation (fixed value: 90%), the TOE stores a saturation audit record and sends a warning email to the authorized administrator. If the audit data storage remains saturated thereafter, the TOE ignores subsequently auditable events.

Related SFR: FAU_STG.5

## 7.2. Cryptographic support

TOE uses the validated cryptographic module specified in '[Table 46] Validated Cryptographic Module Used by the TOE]'.

### 7.2.1. Cryptographic Key Generation

The TOE generates cryptographic keys in accordance with the cryptographic algorithms and key sizes specified in '[Table 38] List of Cryptographic Key Generation Standards]'.
In addition, the TOE generates random numbers required for cryptographic key generation using the Hash_DRBG (SHA-384)-based random bit generator provided by the validated cryptographic module RTJCrypto V1.0.

**[Table 38] List of Cryptographic Key Generation Standards**

| Category | Cryptographic Algorithm | Key Length | Referenced Standard |
|---|---|---|---|
| Integrity verification key | Hash_DRBG (SHA-384) | 256 bit | TTAK.KO-12.0331 |
| DEK (Data Encryption Key) | Hash_DRBG (SHA-384) | 256 bit | TTAK.KO-12.0331 |
| Authentication token encryption key | Hash_DRBG (SHA-384) | 128 bit | TTAK.KO-12.0331 |
| Session key for transmission data encryption | Hash_DRBG (SHA-384) | 128 bit | TTAK.KO-12.0331 |
| SSO server public/private key pair for encryption | RSA | 3072-bit public key | ISO/IEC 18033-2 |
| SSO server public/private key pair for digital signature | RSA | 3072-bit public key | ISO/IEC 18033-2 |
| SSO agent public/private key pair for encryption | RSA | 3072-bit public key | ISO/IEC 18033-2 |
| SSO agent public/private key pair for digital signature | RSA | 3072-bit public key | ISO/IEC 18033-2 |
| KEK (Key Encryption Key) | PBKDF2 (SHA-256) | 256 bit | TTAK.KO-12.0334-Part2 |

Related SFR: FCS_CKM.1

### 7.2.2. Cryptographic Key Distribution

To securely protect the session key used for transmission data encryption between the SSO server and the SSO agent, the TOE distributes the key using RSAES public key encryption provided by the validated cryptographic module. The RSA public key used is that of the peer entity. The SSO server registers the RSA public key of the SSO agent through the Web GUI-based security management interface, and the SSO agent generates its own RSA key pair during installation and includes the RSA public key of the SSO server.
In addition, to establish a secure communication session between the SSO agent and the SSO server, secure session key exchange is performed using the Elliptic Curve Diffie-Hellman (ECDHE) method.

Related SFR: FCS_CKM.2

### 7.2.3. Cryptographic Key Derivation

The SSO server and SSO agent derive the KEK (Key Encryption Key) used to encrypt the DEK (Data Encryption Key) at startup through a password parameter directly entered by the authorized administrator, using the PBKDF2 (SHA-256) key derivation algorithm. Details are specified in '[Table 39] List of Cryptographic Key Derivation Standards]'.

**[Table 39] List of Cryptographic Key Derivation Standards**

| Purpose | TOE Module | Cryptographic Algorithm | Key Length | Referenced Standard |
|---|---|---|---|---|
| KEK(Key Encryption Key) | SSO Server | PBKDF2 (SHA-256) | 256 bit | TTAK.KO-12.0334-Part2 |
| KEK(Key Encryption Key) | SSO Agent | PBKDF2 (SHA-256) | 256 bit | TTAK.KO-12.0334-Part2 |
| TLS v1.3 Session Key | SSO Agent<br>SSO Server | HKDF-SHA256 | 128 bit | RFC8446 |

Related SFR: FCS_CKM.5

### 7.2.4. Timing and Events for Cryptographic Key Destruction

The TOE shall destroy the cryptographic keys and key material listed in "[Table 40] List of Cryptographic Keys" by overwriting them three times with the value '0' when they are no longer required.

Refer to "[Table 41] Timing for Destruction of Cryptographic Keys or Key Material" for the detailed destruction timing.

**[Table 40] List of Cryptographic Keys**

| Category | Cryptographic Algorithm | Key Length | Referenced Standard |
|---|---|---|---|
| KEK (Key Encryption Key) | ARIA/CBC | 256 bit | KS X 1213-1 |
| Integrity Verification Key | HMAC (SHA-256) | 256 bit | ISO/IEC 9797-2 |
| DEK (Data Encryption Key) | ARIA/CBC | 256 bit | KS X 1213-1 |
| Authentication Token Encryption Key | ARIA/CBC | 128 bit | KS X 1213-1 |
| Transmission Information Encryption Session Key | ARIA/CBC | 128 bit | KS X 1213-1 |
| SSO Server Private Key for Encryption | RSAES (SHA-256) | 3072 bit | ISO/IEC 18033-2 |
| SSO Server Private Key for Digital Signature | RSA-PSS (SHA-256) | 3072 bit | ISO/IEC 18033-2 |
| SSO Agent Private Key for Encryption | RSAES (SHA-256) | 3072 bit | ISO/IEC 18033-2 |
| SSO Agent Private Key for Digital Signature | RSA-PSS (SHA-256) | 3072 bit | ISO/IEC 18033-2 |
| TLS v1.3 Session Key | ECDHE | 256 bit | Immediately after use (Memory) |
| TLS v1.3 Session Key | HKDF | 128 bit | Immediately after use (Memory) |
| TLS v1.3 Session Key | AES_128_GCM | 128 bit | Immediately after completion of use (Memory) |

**[Table 41] Timing for Destruction of Cryptographic Keys or Key Material**

| Storage Location | Object to be Destroyed | Timing of Destruction |
|---|---|---|
| Memory | Password for KEK derivation | Immediately after use |
| Memory | KEK (Key Encryption Key) | Immediately after use |
| Memory | Integrity Verification Key | Upon invocation of product termination process |
| Memory | DEK (Data Encryption Key) | Upon invocation of product termination process |
| Memory | DEK Public Security Parameter (IV) | Upon invocation of product termination process |
| Memory | Authentication Token Encryption Key | Immediately after use |
| Memory | Authentication Token Public | Immediately after use |

| | | |
|---|---|---|
| | Security Parameter (IV) | |
| Memory | Transmission Information Encryption Session Key | Immediately after use |
| Memory | Transmission Information Public Security Parameter (IV) | Immediately after use |
| Memory | SSO Server Private Key for Encryption | Upon invocation of product termination process |
| Memory | SSO Server Private Key for Digital Signature | Upon invocation of product termination process |
| Memory | SSO Agent Private Key for Encryption | Upon invocation of product termination process |
| Memory | SSO Agent Private Key for Digital Signature | Upon invocation of product termination process |

Related SFR: FCS_CKM.6

### 7.2.5. Cryptographic Operations

When performing the "FCS_COP.1 Cryptographic Operation" specified in "[Table 42] List of Cryptographic Operation Standards," the TOE shall perform the following cryptographic operations:

- TOE integrity verification: message authentication code cryptographic operation for TOE configuration and executable code.
- KEK generation, TSF data integrity verification, one-way encryption of user password, authentication token verification: hash algorithm cryptographic operation.
- DEK encryption, TSF data encryption, authentication token encryption: block cipher algorithm cryptographic operation.
- Encryption of transmission information encryption session key: RSAES public key cryptographic operation during cryptographic key distribution.
- Authentication token digital signature generation/verification, mutual authentication between SSO Server and SSO Agent: RSA-based digital signature generation/verification cryptographic operation.
- Cryptographic communication between TOE components: confidentiality is maintained through TLS v1.3-based secure communication.
- Cryptographic communication between SSO Server and Mail Server: confidentiality is maintained through TLS v1.3-based secure communication.
- Cryptographic communication for management access between user and TOE: confidentiality is maintained through TLS v1.3-based secure communication.

The TOE shall perform cryptographic operations in compliance with the specified "cryptographic algorithm" and "key length."

**[Table 42] List of Cryptographic Operation Standards**

| Purpose | Cryptographic Algorithm | Key Length | TOE Module | Referenced Standard |
|---|---|---|---|---|
| TOE Integrity Verification | HMAC (SHA-256) | 256 bit | SSO Server<br>SSO Agent | ISO/IEC 9797-2 |
| One-way Encryption of User Password | SHA-256 | 256 bit | SSO Server | ISO/IEC 10118-3 |
| DEK Encryption, TSF Data Encryption | ARIA/CBC | 256 bit | SSO Server<br>SSO Agent | KS X 1213-1 |
| Authentication Token Encryption | ARIA/CBC | 128 bit | SSO Server | KS X 1213-1 |
| Encryption of Transmission Information Session Key | RSAES (SHA-256) | Public Key 3072 bit | SSO Server<br>SSO Agent | ISO/IEC 18033-2 |
| Authentication Token Digital Signature Generation/Verification, Mutual Authentication | RSA-PSS (SHA-256) | Public Key 3072 bit | SSO Server<br>SSO Agent | ISO/IEC 18033-2 |
| TLS v1.3 Secure Communication | TLS_AES_128_GCM (SHA256) | 128 bit | SSO Server<br>SSO Agent | rfc8446<br>ISO/IEC 18033-3 |

Related SFR: FCS_COP.1

### 7.2.6. Random Bit Generation (RBG)

The TOE shall generate random numbers required for cryptographic key generation using the 'Hash_DRBG (SHA-384)' random number generator cryptographic algorithm supported by the validated cryptographic module.

Related SFR: FCS_RBG.1

### 7.2.7. Random Bit Generation (Internal Seeding – Single Source)

The TOE shall use the entropy source provided by the JVM, independent of the underlying operating system, by invoking the JDK internal API 'new SecureRandom().nextBytes()' to supply entropy to the random number generator.

**[Table 43] Detailed Description of Hash_DRBG (SHA-384) Random Number Generator**

**(Unit: bit)**

| Category | Description |
|---|---|
| Block Length | 256 |
| Seed Length | 888 |
| Prediction Resistance | Not supported |
| Reseed Interval | 1000 |
| Minimum Entropy Input | 258 |
| Maximum/Minimum Entropy Input Length | 258 (fixed) |
| Personalization String | Not used |
| Personalization String Length | - |
| Additional Input | Not used |
| Additional Input Length | - |
| Maximum/Minimum Output Length | [24 ~ $2^{19}$] bit range |

Related SFR: FCS_RBG.3

## 7.3. Identification and Authentication

The identification and authentication security functions of the TOE consist of mutual authentication between TOE components for performing security management through the SSO server, user identification and authentication, authentication token generation/verification/destruction, and prevention of authentication data reuse.

### 7.3.1. Handling of Authentication Failures

The TOE locks the corresponding administrator account for 5 minutes when five consecutive failed authentication attempts (fixed value: 5) occur for an authorized administrator and automatically unlocks it after 5 minutes.
The TOE locks the corresponding general user account for 5 minutes when five consecutive failed authentication attempts (fixed value: 5) occur for a general user and automatically unlocks it after 5 minutes.

Related SFR: FIA_AFL.1(1), FIA_AFL.1(2)

### 7.3.2. Mutual Authentication between TOE Components

The TOE performs mutual authentication between the SSO server and the SSO agent when the SSO agent requests a connection to the SSO server using the RSA public/private keys specified in the mutual authentication mechanism below.

**[Table 44] Mutual Authentication Mechanism**

| Category | Cryptographic Algorithm | Key Length | Referenced Standard |
|---|---|---|---|
| Digital Signature Generation/Verification | RSA-PSS (SHA-256) | Public Key 3072bit | ISO/IEC 18033-2 |
| **Mutual Authentication Mechanism** | | | |
| 1. Pre-registration of SSO Agent<br>1) When an authorized administrator registers an SSO agent through the Web GUI-based security management interface, the SSO agent ID and the encryption public key and signature public key generated by the SSO agent for mutual authentication with the SSO server are registered.<br><br>2. Generation of Electronic Signature by SSO Agent<br>1) The SSO agent generates an electronic signature of the following JWT information using its signature private key with the RSAPSS (SHA256) algorithm and transmits it to the SSO server to request mutual authentication. | | | |

- Request Header -> { alg: 'PS256', typ: 'JWT' }
- Request Body -> { timestamp: timestamp, sessionid: session ID, spid: SSO agent ID, nonce: random value }
- Request Signature -> signature (request body)

3. Verification of Electronic Signature by SSO Server
1) The SSO server verifies the received JWT message from the SSO agent using the SSO agent's signature public key with RSAPSS (SHA256), and verifies the SSO agent ID and timestamp expiration time (fixed value: 10 minutes) to authenticate the request.
- Response Header -> { alg: 'PS256', typ: 'JWT' }
- Response Body -> { timestamp: timestamp, sessionid: session ID, spid: SSO agent ID, nonce: random value }
- Response Signature -> signature (response body)

4. Generation of Electronic Signature by SSO Server
1) The SSO server performs authentication and generates an electronic signature of the following JWT information using its private key (RSAPSS), then sends the response message to the SSO agent.
- Request Header -> { alg: 'PS256', typ: 'JWT' }
- Request Body -> { timestamp: timestamp, sessionid: session ID, spid: SSO server ID, nonce: random value }
- Request Signature -> signature (request body)

5. Verification of Electronic Signature by SSO Agent
1) The SSO agent verifies the JWT message received from the SSO server using the SSO server's signature public key with RSAPSS (SHA256), verifies the SSO server ID and timestamp expiration time (fixed value: 10 minutes), and completes mutual authentication.
- Response Header -> { alg: 'PS256', typ: 'JWT' }
- Response Body -> { timestamp: timestamp, sessionid: session ID, spid: SSO server ID, nonce: random value }
- Response Signature -> signature (response body)

Related SFR: FIA_IMA.1

### 7.3.3. Verification of Secrets

The TOE performs password verification for authorized administrators and general users during ID/password-based identification and authentication according to the verification mechanism specified in ' [Table 45] Password Security Criteria Type (1) '.

**[Table 45] Password Security Criteria Type (1)**

| Category | Description |
|---|---|
| Compliance Requirements | Minimum length: 9 characters or more |
| | Maximum length: 255 characters or fewer |
| | Must include at least one digit, one uppercase letter (A–Z), one lowercase letter (a–z), and one special character |
| Prohibited Items | Prohibited from setting the password identical to the user account (ID) |
| | Prohibited from consecutive repetition of identical characters or numbers |
| | Prohibited from sequential input of adjacent keyboard characters or numbers |
| | Prohibited from reusing the previously used password |

Related SFR: FIA_SOS.1

### 7.3.4. Generation of Secrets

After a general user is successfully identified and authenticated, the TOE provides a mechanism for generating an authentication token that meets the following. When generating the authentication token, a unique OTP (random value) and server time information are encrypted.

- Entity responsible for generation/verification of authentication token: SSO server
- Entity responsible for storage of authentication token: PC web browser, SSO server
- Components of authentication token: server ID, user ID, OTP (random value), expiration time (timestamp)
- Cryptographic algorithm for authentication token: ARIA/CBC block cipher using a 128-bit symmetric key
- Integrity algorithm for authentication token: RSA-PSS (SHA-256) digital signature algorithm using a 3072-bit RSA public key

The TOE performs cryptographic operations for authentication token generation using the validated cryptographic module as shown in ' [Table 46] Validated Cryptographic Module Used by the TOE '.

**[Table 46] Validated Cryptographic Module Used by the TOE**

| Module Name | Security Level | Validation Number | Developer | Validation Date | Expiration Date |
|---|---|---|---|---|---|
| RTJCrypto V1.0 | 1 | CM-281-2030.10 | RathonTech Co., Ltd. | October 24, 2025 | October 24, 2030 |

Related SFR: FIA_SOS.2

### 7.3.5. Destruction of Secrets

The TOE destroys the authentication token stored in the SSO server when a general user session terminates or the logout function is invoked. The destruction method overwrites the value with '0' three times.

Related SFR: FIA_SOS.3

### 7.3.6. Timing of Authentication

For authorized administrators, the TOE performs account/password-based authentication and verifies access from pre-approved administrator terminal IP addresses (maximum: 2).

For general users, the TOE performs two-step authentication consisting of account/password-based authentication and single sign-on authentication, verifies access from pre-approved general user IP addresses, and checks additional attributes (SSO agent IP, SSO agent ID) to ensure the business system is accessible to the general user.

Related SFR: FIA_UAU.2

### 7.3.7. Single-use authentication mechanisms

The TOE provides the following functions to prevent reuse of authentication information.

a) Prevention of reuse of administrator authentication information

1) Authorized administrators can access the Web GUI-based security management interface only through the login page.

2) To prevent CSRF (Cross-Site Request Forgery) attacks, a CSRF token is assigned to each page before administrator authentication.

3) The CSRF token assigned during authentication is transmitted with login credentials to prevent reuse.

b) Prevention of reuse of general user ID/password authentication information

1) To prevent CSRF attacks, a nonce is assigned to each page before user authentication.

2) Access is restricted if the assigned nonce is not included in the request.

3) The user authenticates through the SSO agent using the CSRF token issued by the server.

4) This token includes one-time authentication data such as timestamp, nonce, and session ID, is not stored for reuse, and is immediately destroyed after use.

c) Prevention of reuse of general user authentication token

1) After successful initial authentication, the issued authentication token is submitted to the SSO server for verification.

2) The SSO server verifies the OTP (random value) within the authentication token to determine whether the token has been reused and prevents reuse.

Related SFR: FIA_UAU.4(1), FIA_UAU.4(2)

### 7.3.8. Authentication Feedback Protection

During authentication, the TOE displays entered passwords as "▪" instead of the actual characters and does not provide specific failure reasons upon authentication failure. Instead, it provides the message "Authentication failed.", making it difficult for third parties to determine the cause of authentication failure.

Related SFR: FIA_UAU.7

### 7.3.9. Timing of Identification

Before performing any action on behalf of a user, the TOE identifies the user using a unique user ID through account/password-based identification and authentication.

Related SFR: FIA_UID.2

## 7.4. Security Management

The TOE provides authorized administrators with the capability to manage all security functions and restricts administrative roles so that no level other than super administrator exists.

### 7.4.1. Management of Security Functions

The TOE provides a Web GUI-based security management interface for the management actions of the functions listed in [Table 47] Security Management Functions and restricts the use of such security management functions to authorized administrators (super administrators) only.

**[Table 47] Security Management Functions**

| Subcategory | Security Management |
|---|---|
| Identification and Authentication | User registration, deletion, modification, authorization |
| Security Management | Registration, modification, deletion of IP addresses of management terminals |
| | SSO agent inquiry – status, version, applied security policy |
| | SSO agent security policy management – policy configuration |
| Self-Protection | Perform integrity verification of TOE configuration values and the TOE itself upon administrator request |
| Update Protection | Inquiry of TOE version information |
| Audit Records | Inquiry of audit records |
| FPT_TST.1 | Execution of TOE configuration value verification and TOE self-integrity check upon administrator request |

Related SFR: FMT_MOF.1

### 7.4.2. Management of TSF data

The TOE restricts the use of TSF data management functions, such as registration, modification, deletion, and inquiry of TSF data listed in ' [Table 48] TSF Data List ', so that only authorized administrators can use them through a Web GUI-based security management interface.

**[Table 48] TSF Data List**

| Related SFR | TSF Data Management Actions |
|---|---|
| FIA_UAU.1<br>FIA_UID.1 | Authorization assignment to user accounts (ID) |

| | |
|---|---|
| FIA_UAU.1<br>FIA_UID.1 | Addition and deletion of user IDs |
| FMT_MTD.1<br>FMT_PWD.1 | Addition, deletion, and modification of user passwords |
| FTA_TSE.1 | Registration, modification, deletion of management terminal IP addresses |
| FMT_MTD.1 | Agent inquiry - Mandatory inquiry information: agent version, applied security policy, agent operational status (enabled/disabled), agent integrity verification result (success/failure) |
| FMT_MTD.1 | Agent security policy management |
| FMT_MTD.1 | Configuration of authentication information for access by external IT entities |
| FMT_MTD.1 | Inquiry of identification information of the TOE and TOE components (e.g., server, agent) |
| FAU_SAR.1 | Inquiry of audit records |
| FPT_TST.1 | Execution of TOE configuration value verification and TOE self-integrity check upon administrator request |

Related SFR: FMT_MTD.1

### 7.4.3. ID and Password Management

The TOE provides functions to create and change passwords for general users and administrators. Password creation is permitted only to authorized administrators (super administrators) and can be performed through the Web GUI-based administrator interface of the SSO server. Password modification can be performed by authorized administrators (super administrators) for all accounts, and general users may change their own passwords only when necessary. The application of password verification rules during password creation or modification shall refer to the FIA_SOS.1 component specified in "6. Security Requirements" of this Security Target.

In addition, when an authorized administrator initially accesses the Web GUI-based administrator interface, the TOE forcibly redirects the administrator to a password change page to require password modification.
The ID and password composition rules for authorized administrators are as shown in the following tables.

**[Table 49] Password Function List**

| No. | Function | Remarks |
|---|---|---|
| 1 | Password change upon initial administrator login | |
| 2 | Password change upon administrator information modification | |
| 3 | General user password creation/change | |

**[Table 50] Password Security Criteria Type (1)**

| Category | Description |
|---|---|
| Compliance Requirements | Minimum length: 9 characters or more |
| | Maximum length: 255 characters or fewer |
| | Must include at least one digit, one uppercase letter (A–Z), one lowercase letter (a–z), and one special character |
| Prohibited Items | Prohibited from setting the password identical to the user account (ID) |
| | Prohibited from consecutive repetition of identical characters or numbers |
| | Prohibited from sequential input of adjacent keyboard characters or numbers |
| | Prohibited from reusing the previously used password |

**[Table 51] ID Function List**

| No. | Function | Remarks |
|---|---|---|
| 1 | General user ID creation | |

**[Table 52] ID Composition Rules and Length**

| Category | Description |
|---|---|
| **ID Composition Rules** | ID Composition Rules  Uppercase and lowercase letters (A-Z, a-z), digits (0-9) |
| | Prohibition of spaces and control characters (e.g., tab, newline) |
| | Restriction on consecutive identical characters (e.g., prevention of aaa, 1111) |
| | Restriction on common words to prevent dictionary attacks<br>: root, admin, user, test, manager |
| **ID Length Criteria** | Minimum length: 3 characters |
| | Maximum length: 50 characters |
| **Security Requirements** | Uniqueness: prevention of duplicate IDs within the same system |
| | Prohibition of ID identical to password: if the ID and password are identical, it is not permitted |

Related SFR: FMT_PWD.1

### 7.4.4. Specification of Management Functions

The TOE provides a Web GUI-based administrator interface that enables authorized administrators to perform the management functions listed under FMT_MOF.1 (management of security functions), FMT_MTD.1 (management of TSF data), and FMT_PWD.1 (extended) (ID and password management), as specified in "7. TOE Summary Specification."

Related SFR: FMT_SMF.1

### 7.4.5. Security roles

The TOE provides only the role of authorized administrator as shown in [Table 53] User Classification and Roles.

**[Table 53] User Classification and Roles**

| User Category | Role | Description |
|---|---|---|
| Authorized Administrator | Super Administrator | Authorized administrator with all permissions (Read/Write) |
| General User | User | General user who can change their own password |

Related SFR: FMT_SMR.1

## 7.5. Protection of the TSF

The TOE provides the following functions for TSF protection.

### 7.5.1. Maintenance of a secure state upon failure

If an error occurs in the entropy source, such as a failure in the noise source integrity test, the TOE transitions to a critical error state, blocking the operation of the validated cryptographic module and the TOE. An authorized administrator then maintains a secure state by reinstalling the damaged file or restarting the WAS server, as described in the operating manual.

Related SFR: FPT_FLS.1

### 7.5.2. Basic internal TSF data transfer protection

When TSF data is transmitted between separate parts of the TOE, the cryptographic communication method uses the TLS v1.3 encryption communication protocol to provide a secure channel for protecting TSF data between TOE components. TLS v1.3 ensures data integrity and confidentiality, protecting data from eavesdropping and tampering that may occur during communication. The table below details the currently used cryptographic communication protocol.

**[Table 54] Specification of Cryptographic Communication Standard Protocols**

| Category | Description | Reference Standard |
|---|---|---|
| **Encrypted Communication Protocol** | TLS | rfc8446 |
| **Version** | 1.3 | rfc8446 |
| **cipher suites** | TLS_AES_128_GCM_SHA256 | rfc8446 |
| **Confidentiality (Encryption)** | AES_128 | rfc3826 |
| **Integrity (Authentication)** | GCM | rfc5288 |
| **Key Exchange Algorithm** | Ephemeral Diffie-Hellman or Elliptic Curve Diffie-Hellman Ephemeral | rfc9528 rfc8037 |

Related SFR: FPT_ITT.1

### 7.5.3. Basic protection of stored TSF data

The TOE performs encryption on the TSF data listed in the table below to protect stored TSF data from unauthorized exposure and modification. It provides functionality to generate the KEK used to encrypt the DEK during SSO server and SSO agent startup, and then encrypts the DEK using the KEK. Furthermore, the KEK key used within the TOE is generated via a password-based key derivation scheme, employing the PBKDF2 password-based key derivation function defined in PKCS#5. The password used is the one directly entered by an authorized administrator during SSO server or SSO agent startup. The list of TSF data subject to protection and the applied cryptographic algorithms are as shown in the table below.

**[Table 55] Cryptographic Algorithms Applied to TSF Data Protection**

| Category | TSF Data | Encryption Key | Cryptographic Algorithm | Storage Location |
|---|---|---|---|---|
| **SSO Server** | Administrator password | - | SHA-256 | DBMS |
| | General user password | - | SHA-256 | DBMS |
| | DBMS connection information | DEK | ARIA/CBC/256 | File |
| | Mail server connection information | DEK | ARIA/CBC/256 | File |
| | Configuration information | DEK | ARIA/CBC/256 | DBMS |
| | Authentication token | Key for authentication token encryption | ARIA/CBC/128 | Memory |
| | Server private key for encryption | DEK | ARIA/CBC/256 | File |
| | Server private key for digital signature | DEK | ARIA/CBC/256 | File |
| | Key for authentication token encryption | Server public key for encryption | ARIA/CBC/128 | Memory |
| | Session key for transmission data encryption | Agent public key for encryption | ARIA/CBC/128 | Memory |
| | Key for integrity verification | DEK | ARIA/CBC/256 | File |
| | DEK (Data Encryption Key) | KEK | ARIA/CBC/256 | File |
| | Integrity value | Key for integrity verification | HMAC (SHA-256) | File |
| **SSO Agent** | Configuration information | DEK | ARIA/CBC/256 | File |
| | Agent private key for encryption | DEK | ARIA/CBC/256 | File |
| | Agent private key for digital signature | DEK | ARIA/CBC/256 | File |
| | Session key for transmission | Server public key for | ARIA/CBC/128 | Memory |

| | | | | |
|---|---|---|---|---|
| | data encryption | encryption | | |
| | Key for integrity verification | DEK | ARIA/CBC/256 | File |
| | DEK (Data Encryption Key) | KEK | ARIA/CBC/256 | File |
| | Integrity value | Key for integrity verification | HMAC (SHA-256) | File |

Related SFR: FPT_PST.1

### 7.5.4. TSF testing

The TOE performs a self-test ('[Table 56] List of self-tests executed by the TSF ') upon startup of the SSO server or SSO agent, and periodically performs self-tests every 12 hours after startup. Additionally, it provides a function to notify authorized administrators via email in real-time when a self-test fails.

**[Table 56] List of self-tests executed by the TSF**

| Category | Self-Test List | Remarks |
|---|---|---|
| SSO Server | SSO Server Process Test | |
| | SSO Server Integrity Verification | |
| | SSO Server Cryptographic Module Self-Test | Including Noise Source Integrity Test |
| SSO Agent | SSO Agent Process Test | |
| | SSO Agent Integrity Verification | |
| | SSO Agent Cryptographic Module Self-Test | Including Noise Source Integrity Test |

#### 7.5.4.1. Integrity verification

The integrity verification function checks whether TSF data and TSF executable files have been tampered with and is performed on the SSO server or SSO agent by an authorized administrator.
Additionally, an authorized administrator can manually perform SSO server integrity verification by logging into the Web GUI-based security management interface and clicking the integrity verification menu.
During integrity verification, the HMAC-SHA256 cryptographic algorithm is used to validate data validity by comparing the stored HMAC value with the newly generated HMAC value.

The integrity verification items performed by the TOE are as follows:

| Category | Self-Test List | Details |
|---|---|---|
| SSO Server | {WAS_HOME}/webapps/{Server_HOME}/WEB-INF<br>Example) /home/rathon/apache-tomcat-11.0.18/webapps/rathonssoserver/WEB-INF<br>{WAS_HOME}/lib/RathonSSO_Crypto_Kit-4.0.1.jar<br>{WAS_HOME}/lib/rt-jcrypto-1.0.jar<br>{WAS_HOME}/conf/server.xml<br>{WAS_HOME}/bin/setenv.sh | SSO Server Configuration File, JAR Library (All) |
| SSO Agent | {WAS_HOME}/webapps/{Agent_HOME}/WEB-INF<br>Example) /home/rathon/apache-tomcat-11.0.18/webapps/rathonssoagent/WEB-INF<br>{WAS_HOME}/lib/RathonSSO_Crypto_Kit-4.0.1.jar<br>{WAS_HOME}/lib/rt-jcrypto-1.0.jar<br>{WAS_HOME}/bin/setenv.sh | SSO Agent Configuration Files, JAR Libraries (All) |

- WAS_HOME : WAS Server Home Path
- Server_HOME : SSO Server Home Path
- Agent_HOME : SSO Agent Home Path

The integrity verification conditions for TOE are as follows:

| Category | Integrity Verification Condition |
|---|---|
| SSO Server | Performs integrity verification periodically every 12 hours during startup and normal operation. |
| SSO Server | Performs integrity verification manually upon request by an authorized administrator. |
| SSO Agent | Performs periodic integrity checks every 12 hours during startup and normal operation. |

Related SFR: FPT_TST.1

## 7.6. TOE access

### 7.6.1. Per user attribute limitation on multiple concurrent sessions

The TOE limits the maximum number of concurrent sessions for users with the same privilege level and for the same user to 1. When a concurrent login attempt occurs, the newly attempted session is denied. The session lock is released after a fixed period of 5 minutes.

Related SFR: FTA_MCS.2

### 7.6.2. TSF-initiated termination

For authorized administrators and general users, the TOE determines inactivity when no actions such as clicking or refreshing the Web GUI screen occur after successful login. If the inactivity period reaches the fixed value of ten (10) minutes, the TOE compares the login session time with the SSO server time, and if more than 10 minutes have elapsed, the session is terminated.

Related SFR: FTA_SSL.3

### 7.6.3. TOE session establishment

Prior to establishing a management access session for an authorized administrator of the SSO server, the TOE provides a management access session restriction function based on permitted IP address information. The number of permitted IP addresses provided by the TOE is fixed at two (2), and up to two IP addresses may be registered, deleted, or modified by an authorized administrator with the super administrator role. Characters specifying IP address ranges are not allowed.
In addition, once the TOE is properly installed, authorized administrators may access the TOE Web GUI-based security management interface through a web browser on the administrator PC.
The TOE allows access to the Web GUI-based security management interface (HTTPS) only after the authorized administrator has successfully completed identification and authentication.

Related SFR: FTA_TSE.1

## 7.7. Trusted path/channels

### 7.7.1. Inter-TSF trusted channel

The TOE provides a secure channel between TSFs using the TLS v1.3 encrypted communication protocol.
TLS v1.3 ensures the integrity and confidentiality of data in transit, thereby protecting data from eavesdropping and tampering.
TSF components cooperate to establish a secure communication path when initiating communication. Trusted IT products also use TLS v1.3 to encrypt data, thereby ensuring a secure network environment and preventing security threats that may occur during the transmission of management information.

- Confidentiality and Integrity Assurance: AES-128, a symmetric-key encryption algorithm, is used to maintain the confidentiality of transmitted data. Simultaneously, data integrity is secured through the GCM (Galois/Counter Mode) authentication method, preventing tampering and forgery.

Related SFR: FTP_ITC.1