

Voltage SecureData Appliance 7.0.2 Security Target

Date: September 5, 2025
Version: 0.19
Prepared By: Dawn Adams
Prepared For: OpenText
275 Frank Tompa Drive
Waterloo ON N2L 0A1
Canada

Abstract

This document provides the basis for an evaluation of a specific Target of Evaluation (TOE), Voltage SecureData Appliance 7.0.2. This Security Target (ST) defines a set of assumptions about the aspects of the environment, a list of threats that the product intends to counter, a set of security objectives, a set of security requirements and the IT security functions provided by the TOE which meet the set of requirements.

Contents

Contents.....	2
1 Introduction	4
1.1.1 ST Reference.....	4
1.2 TOE Reference	4
1.3 Document Conventions	4
1.4 Document Terminology.....	5
1.5 TOE Overview	5
1.5.1 Hardware and Software Supplied by the IT Environment	6
1.6 TOE Description	7
1.6.1 Physical Boundary	8
1.6.2 Logical Boundary	13
2 Conformance Claims	15
2.1 CC Conformance Claim	15
2.2 PP Claim	15
2.3 Package Claim	15
2.4 Conformance Rationale	15
3 Security Problem Definition	16
3.1 Threats.....	16
3.2 Assumptions	17
4 Security Objectives.....	18
4.1 Security Objectives for the TOE.....	18
4.2 Security Objectives for the Operational Environment	18
4.3 Security Objectives Rationale	19
4.4 Rationale for TOE Threats and Assumptions.....	20
5 Extended Components Definition.....	21
6 Security Requirements.....	22
6.1 Security Functional requirements	22
6.1.1 Security Audit (FAU)	22
6.1.2 Cryptographic Support (FCS)	23
6.1.3 User Data Protection (FDP)	24

6.1.4	Identification and Authentication (FIA).....	26
6.1.5	Security Management (FMT).....	27
6.1.6	TOE ACCESS (FTA).....	28
6.1.7	Trusted Path/Channels (FTP).....	28
6.2	Security Assurance Requirements Rationale.....	29
6.3	Dependency Rationale.....	30
6.4	Objectives to SFR Mapping.....	32
6.5	Objectives to SFRs Rationale	33
7	TOE Summary Specification	34
7.1	TOE Security Functions.....	34
7.2	Security Audit	34
7.3	Cryptographic Support	35
7.4	User Data Protection	36
7.4.1	Identity Authorization	36
7.4.2	IP Authorization.....	37
7.5	Identification and Authorization	38
7.5.1	Administrator I&A	38
7.5.2	Client I&A	39
7.6	Security Management	39
7.7	TOE Access.....	40
7.8	Trusted Path/Channels	41

1 Introduction

This section identifies the Security Target (ST), Target of Evaluation (TOE), Security Target organization, document conventions, and terminology. It also includes an overview of the evaluated product.

1.1.1 ST Reference

ST Title	Voltage SecureData Appliance 7.0.2
ST Revision	0.19
ST Publication Date	September 5, 2025
Author	Dawn Adams

1.2 TOE Reference

TOE Reference	Voltage SecureData Appliance 7.0.2
	Note: The TOE, or SecureData, or SDA are used interchangeably in this ST, and all refer to Voltage SecureData Appliance 7.0.2

SECTION	TITLE	DESCRIPTION
1	Introduction	Provides an overview of the TOE and defines the hardware and software that make up the TOE as well as the physical and logical boundaries of the TOE
2	Conformance Claims	Lists evaluation conformance to Common Criteria versions, Protection Profiles, or Packages where applicable
3	Security Problem Definition	Specifies the threats, assumptions and organizational security policies that affect the TOE
4	Security Objectives	Defines the security objectives for the TOE/operational environment and provides a rationale to demonstrate that the security objectives satisfy the threats
5	Extended Components Definition	Describes extended components of the evaluation (if any)
6	Security Requirements	Contains the functional and assurance requirements for this TOE
7	TOE Summary Specification	Identifies the IT security functions provided by the TOE and identifies the assurance measures targeted to meet the assurance requirements.

Table 1 – ST Organization and Section Descriptions

1.3 Document Conventions

The notation, formatting, and conventions used in this Security Target are consistent with those used in Version 3.1 of the Common Criteria. Selected presentation choices are discussed here to aid the Security Target reader. The Common Criteria allows several operations to be performed on functional requirements: The allowable operations defined in Part 2 of the Common Criteria are *refinement*, *selection*, *assignment* and *iteration*.

- The assignment operation is used to assign a specific value to an unspecified parameter, such as the length of a password. An assignment operation is indicated by showing the value in square brackets, i.e. [assignment_value(s)].

- The refinement operation is used to add detail to a requirement, and thus further restricts a requirement. Refinement of security requirements is denoted by bold text. Any text removed is indicated with a bolded strikethrough format (Example: **TSF** or **~~TSF~~**).
- The selection operation is picking one or more items from a list in order to narrow the scope of a component element. Selections are denoted by italicized text.
- Iterated functional and assurance requirements are given unique identifiers by appending to the base requirement identifier from the Common Criteria an iteration number inside parenthesis, for example, FMT_MTD.1.1 (1) and FMT_MTD.1.1 (2) refer to separate instances of the FMT_MTD.1 security functional requirement component.

When not embedded in a Security Functional Requirement, italicized text is used for both official document titles and text meant to be emphasized more than plain text.

1.4 Document Terminology

The following table describes the acronyms used in this document:

TERM	DEFINITION
CC	Common Criteria version 3.1
EAL	Evaluation Assurance Level
EOE	Events Originating External to the TOE
ISO	International Standards Organization. When referring to a CD or DVD it means ISO-9660
NTP	Network Time Protocol
OSP	Organizational Security Policy
OVF	Open Virtualization Format
SFP	Security Function Policy
SFR	Security Functional Requirement
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Function

Table 2 – Acronyms Used in Security Target

1.5 TOE Overview

The TOE, Voltage SecureData Appliance 7.0.2 (SDA), provides protection of sensitive data, such as credit card numbers and Social Security numbers, stored in databases and applications. It enables enterprises to ensure that sensitive data residing in databases and used in applications is protected as it is collected, used, stored, and distributed to less controlled environments. SDA provides the ability to implement a comprehensive solution for data protection offering data de-identification, data masking, and data redaction that requires minimal changes to the underlying systems. The Simple API provides a set of functions that are callable from existing C, C#/.NET, and Java applications, enabling data protection functionality to be included into any such application and applications to communicate with the SDA to obtain keys. The SDA can be used in a single server deployment or in a multiple server deployment. Only the single server deployment is being evaluated.

The TOE provides:

- Cryptographic algorithms to calling APIs
- Communications protected by TLS v1.2

- Administrator management of TOE security functions
- Enforced access control to the TOE
- Auditing of events
- Identification and authorization of operators
- User terminations of sessions and TOE terminations of inactive sessions
- TLS v1.2 protected connections with TOE

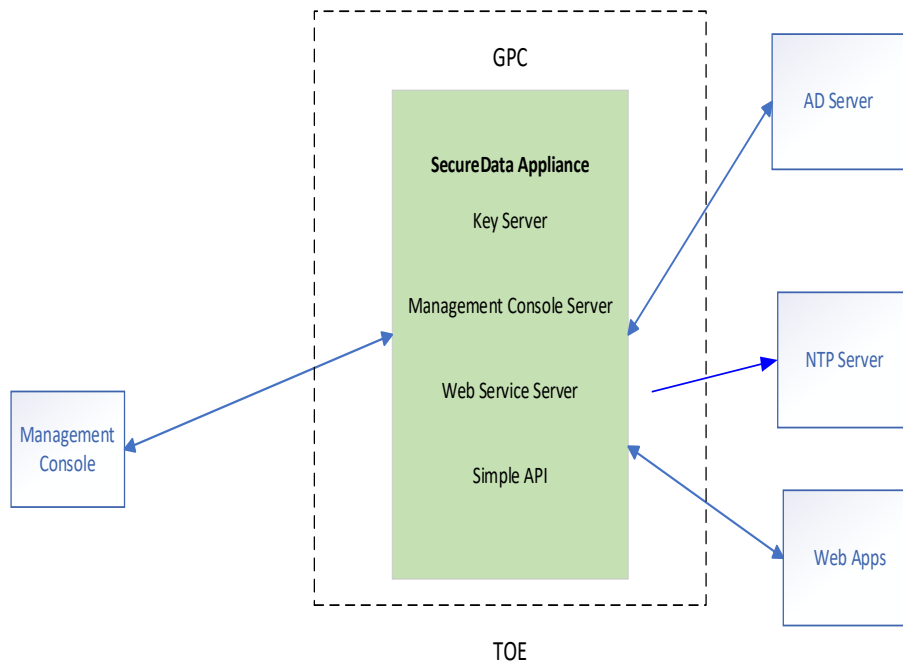


Figure 1 - TOE Evaluated Configuration

The TOE is Green. The White is not in the TOE.

1.5.1 Hardware and Software Supplied by the IT Environment

Voltage SecureData Appliance software installation requires the following hardware:

- At least 3 Ghz x86 64-bit architecture processor, with 8 CPU cores
- At least 8 GB RAM
- At least 80 GB disk space. Recommended disk size for Management Console Hosts is 256 GB.
- If you are installing on hardware, use a DVD drive (internal or external).
- 1GB Ethernet Network Interface Card (NIC)
- A hardened Linux-based operating system is loaded from the Voltage-provided ISO image file.

The following hardware is supported:

- All hardware compatible with SUSE Linux Enterprise 15, 64 bit¹
- VMware ESXi Server 7.0 VMware ESXi Server 7.0 Installing the Voltage SecureData Appliance Software

¹ OpenText supplies openSUSE Linux 15, 64 bit with the SDA ISO.

NOTE: The default log size is now 100MB.

NOTE: Versions of VMware ESXi prior to 6.0 might work correctly, but only versions 7.0 has been validated.

The operational environment requires:

NTP server is required to provide reliable timestamps to the TOE.

LDAP server is required to provide authentication.

1.6 TOE Description

The TOE is a software product that provides protection of sensitive data such as credit card numbers and social security numbers that are stored in databases and applications. It enables enterprises to ensure that sensitive data residing in databases and used in applications is protected as it is collected, used, stored, and distributed to less controlled environments. SDA provides the ability to implement a comprehensive solution for data protection offering data de-identification, data masking, and data redaction that requires minimal changes to the underlying systems. The Simple API provides a set of functions that are callable from existing C, C#/.NET, and Java applications, enabling data protection functionality to be included into any such application and applications to communicate with the SDA to obtain keys.

The TOE provides the following capabilities that external applications can use for data protection and masking:

- Format Preserving Encryption (FPE)—encrypts data so that the ciphertext has the same length and character set as the input data
- embedded Format Preserving Encryption (eFPE)—encrypts data so that identity information is embedded into the ciphertext, preserving the length but not the character set of the input data
- tokenization—a feature that allows the original data to be replaced by an alias or “token”. In tokenization, the system uses tokens instead of actual account numbers. Tokens are generated based on a credit card format or SSN format, and created using a random number generator. Tokens have no mathematical relationship with the live data.
- Identity-Based Encryption (IBE)—an asymmetric algorithm that encrypts data without preserving its length or character set
- Identity-Based Symmetric Encryption (IBSE)—a non-format-preserving symmetric encryption algorithm. The output of this protection method is a single PKCS7 blob that contains meta-data in addition to the protected data, so it does not preserve the format of the original data. The meta-data consists of the fully qualified identity that was used for protection and an optional random tweak value that is generated at protection time.

The TOE provides centralized key management and tools for performing the actual data masking.

All the authentication, key management, and operational complexity can be abstracted into a Web Service call or an API call. Web-based data access can also be triggered from a database stored procedure, facilitating data-driven access for cases where applications cannot change at all. All integration options are available and can be mixed and matched as needed. Protection and access operations can handle bulk or single data protection operations.

When configuring the system, a master secret for each algorithm used (for example, one for FPE and one for IBE) is created. Individual protection keys are mathematically derived from that master secret. Because only one master secret per algorithm is created and the rest of the keys are derived, the TOE requires only a one-time backup, and eliminates the need to persistently store individual protection keys. The TOE also supports storage of root keys in a Hardware Security Module (HSM). The TOE supports both Atalla and Thales nShield Connect HSMs, but both types cannot be used simultaneously with a TOE configuration under the control of a single Management Console.

The functionality provided by the TOE is accessed using the SecureData Simple API client software, which provides an interface that supports encryption within an application. The SecureData Simple API provides support for applications developed in C, Java, and C#/.NET.

The following clients are also supported by SDA, but have not been included within the scope of the evaluation:

- SecureData File Processor—supports protection of data within a CSV file or within a text file with fixed-width columns.
SecureData for Teradata—provides user-defined functions (UDFs) that protect and access data stored in a Teradata database.
- SecureData z/Protect - provides interoperable support with z/OS systems, including ASCII-EBCDIC transparency.

1.6.1 Physical Boundary

The TOE SDA 7.0.2 is comprised of

- Management Console Server
- Key Server
- Web Service Server
- Simple API v 6.22.02

The SDA also includes the following components that are outside the TOE boundary:

- Web Front End Server (FES)—the FES supports Voltage SecureData Web, a scalable and reliable data-protection solution that uses the Page Integrated Encryption (PIE) protocol. This protocol lets eCommerce merchants protect Primary Account Number (PAN) data and other data strings exchanged in web-based transactions. Voltage SecureData Web is not included within the scope of the evaluation.
- KMS for Hadoop TDE—the SDA can be configured to be used as the Key Management Server (KMS) for Hadoop Transparent Data Encryption (TDE), replacing the native Hadoop KMS. Hadoop TDE uses keys generated by the KMS when automatically encrypting and decrypting files in specified directories within a Hadoop Distributed File System (HDFS). Voltage SecureData for Hadoop is not included within the scope of the evaluation.

1.6.1.1 Key Management Server

The Key Management Server supports centralized SecureData key management. SecureData is built around a centralized key management system that coordinates the generation and issuance of FPE keys, AES keys, and IBE keys. Unlike traditional systems using randomly generated keys that require

complex backup and recovery procedures, the Key Management Server provides stateless key generation through the use of a Key Derivation Function (KDF).

The Key Management Server also provides an authentication system that can integrate with any existing credential store. For example, an existing LDAP or Active Directory can be leveraged to provide authentication of applications, users, or machines, including dynamic group-based authentication. Multiple authentication methods can be utilized, and authentication settings can be changed over time as requirements evolve.

1.6.1.2 *Management Console Server*

The Management Console provides a web-based interface to the Management Console Server to support centralized configuration and reporting across the SecureData solution. Using the Management Console, administrators can:

- Define the Key Management policy, which defines attributes of the Key Management Server
- Define attributes for Web Service API access, including authorizations and required authentication information
- Define the type of authentication that must be used by anyone requesting keys from the Key Management Server
- Define formats for all data types including credit cards, US Social Security numbers, regular numbers, dates, and variable-length and specified-format strings
- Manage mask settings for access using the SecureData Web Service APIs
- Manage TLS parameters and credentials
- Monitor events that happen on one or more Appliances
- Control the type of access required for the network and specific configuration actions
- Perform backup and restore procedures
- Define additional Appliance administrators.

Only one Management Console can be active at any particular time for a SecureData deployment. Regardless of how the enterprise is using the SecureData data protection products, they must be configured through the Management Console.

1.6.1.3 *Web Service Server*

The Web Service Server provides a data protection interface that can be used by web applications capable of consuming Web Services Description Language (WSDL) information. Web Services are an industry standard method of integrating applications with external services. Web implementations are available in a diverse set of application platforms from web browsers to mainframes. The Web Service allows practically any application to make use of the functionality in SecureData.

The Web Service provides an API for protecting and accessing data. It also provides specialized operations for protecting and accessing commonly used data formats, such as credit card numbers and Social Security numbers, and it provides array interfaces to FPE calls in order to optimize the performance of batch operations.

1.6.1.4 *Simple API*

The Voltage SecureData solution is designed to provide protection of data, including credit card numbers, U.S. Social Security numbers, and other data stored in databases and applications. The Simple API allows you to include the following types of cryptographic operations in your application. The TOE provides its security functionality through implementation and management of the following features:

- Identities
- Districts
- Keys
- Formats
- Masked Access
- Tweaking.

Identities

The SecureData client software provides an abstraction that allows client applications to protect and access data based on key names or application names. Client applications do not need to store keys; instead they use a key name to refer to a related set of data. Internally, the Key Management Server component derives the key based on the key name, and uses the derived key to protect or access the data.

In the SecureData APIs and CL, the key name is referred to as the ***identity***. By presenting the credentials for the identity, the client application is permitted, as that identity, to operate on a given piece of data. The use of identities permits seamless integration of data protection with authentication.

The TOE specifies an identity in the format of an email address (e.g., pci@example.com). Note that even though the identity is specified as an email address, it does not necessarily need to refer to a valid mailbox.

Districts

The district is an entity created through the Management Console and contains the system parameters set by the administrator, in addition to district policy and root certificates.

An identity does not become a public key until it is combined with a parameter set, which comes from a district. District parameters must be requested by each client application by downloading the <https://voltage-pp-0000.<domain>/policy/clientPolicy.xml> file.

A domain name is typically used when creating a district. Thenceforward, each time a new district is created, the same domain name is used with a new serial number. The administrator must configure a TLS certificate for the domain name used in the district. If the domain name for a district is changed, any data protected using the previous district name cannot be accessed, because the key is derived from the identity and the district parameters.

Keys

When a client program makes a call to the TOE for data protection, it must provide the following parameters:

- Identity
- Authentication method
- Authentication credentials.

When it receives this information, the data protection services of the TOE issue a key request on behalf of the client program to the Key Management Server. Based on the common name identity, the Key Management Server checks its rules to determine how to authenticate the request. If the authentication succeeds, the key is returned to the originator of the request. If not, the request is rejected and the TOE returns an error to the client program.

When processing authentication methods, the Key Management Server does not consider the order of the matching functions. The TOE determines which authentication methods are applicable for a given key request as follows:

- Match the IP address of the key request against the IP address patterns of the authentication methods
- Match the identity of the key request against the identity patterns of the authentication methods
- Match the type of authentication in the key request to the type of authentication method specified. For example, shared secret key requests can only be handled by Shared Secret authentication methods, and username/password key requests can only be handled by LDAP authentication methods (or custom plug-in methods).

It does not matter which method is tested first because if it fails, the next method is tested until a match is found. If all methods have failed then the authentication request fails.

In the case of the API, a key request is made when the encryption or decryption operation is requested rather than when the encryption object is created. When an application needs to retrieve a key for encryption or decryption, it makes an API call that includes the application's identity in the call parameters via the context object.

The TOE provides administrators the capability to define the following types of data format that can then be used by SecureData clients:

- Credit card number formats
- US Social Security Number formats
- A variable-length string format with input and output alphabets set (such as all ASCII alphabetic characters).
- A specified-format string that defines a fixed-length pattern of characters, such as "DDDD-CCCC".
- A number, which is protected to another number within a specified range. Note that the protected value might have a different number of digits than the plaintext value.
- A date, which is protected to another date within a specified range.

Formats are used to:

- Embed information about the key used to protect the data in the protected data itself. This can be used for credit card numbers, social security numbers, and variable length string formats. To

allow the additional information to be included in the data while keeping the length constant, the set of characters allowed in the output must be larger than the set of characters allowed in the input.

- Specify the use of database-driven tokenization, where plaintexts and pointers to the data are stored in a separate protected database. Database-driven tokenization is available for all formats.
- Specify the use of Secure Stateless Tokenization™ (SST) technology, where a set of metadata is used for tokenization operations, rather than a separate database. SST is available for credit card number formats and US social security number formats only.
- Specify leading and trailing digit protection and the use of short data lengths for Credit Card formats.

Masked Access

Masked access is supported by the Web Services Server and allows administrators to set different masking rules for different identities. Masking rules allow specific users to see only portions of the data. The remaining portion of the data display is substituted with a specified character, such as “X” or “*”, instead of the actual characters. For example, a configuration could display the last four digits of a credit card number and mask the initial numbers with X as the mask character, such as XXXX XXXX XXXX 1234.

Masked data is useful if some users are not authorized to see fully accessed data. It allows a client program to display only allowed portions of the accessed data while continuing to protect the portions of the data that those users are not authorized to view.

Tweaking

Usually, encrypting multiple instances of the same data with the same key produces the same protected value for each of those instances. When a large amount of data is encrypted and some of the values are repeated, the encrypted values are also repeated. These repeated values could potentially provide information about the plaintext values from which they were derived.

The TOE addresses this issue by providing the capability to apply a tweak to each encryption operation that makes the output of each operation unique. A random value is added during data protection so that the same input value protected multiple times with the same key can produce different protected results. Although tweaking increases the time it takes to protect data, it provides added security when protecting a set of data that includes repeated values.

1.6.1.5 TOE Product Documentation

The TOE Product Documentation is not publicly available. Once the TOE has been purchased and a client account has been established, the following documentation is released to the client along with the software on the software download page:

SecureData Appliance, Software Version 7.0.2, Administrator Guide, April 2024

SecureData Appliance, Software Version 7.0.2 Installation Guide, April 2024

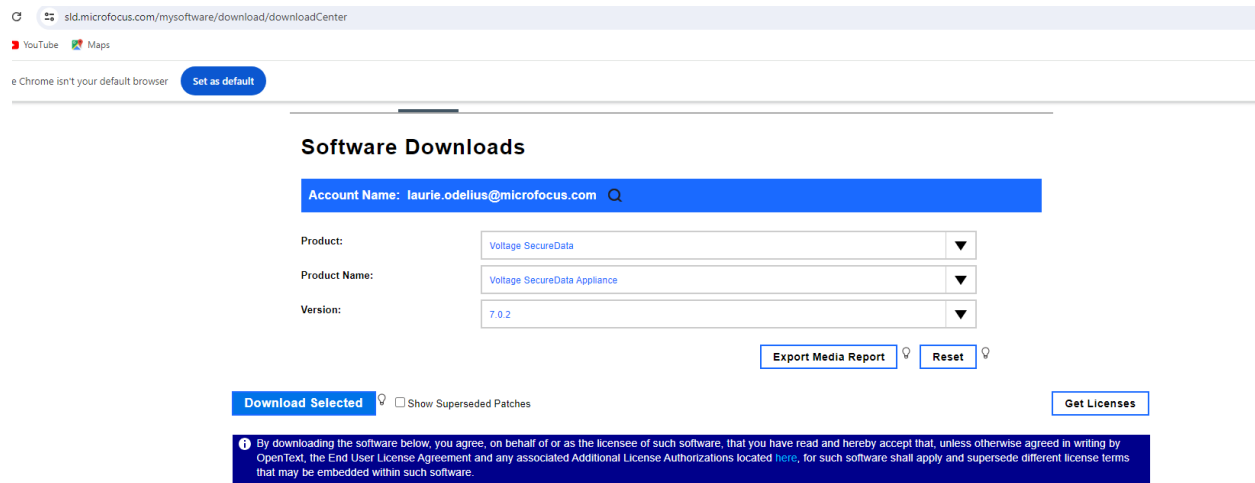
A Common Criteria supplement, Voltage Secure Data Appliance 7.0.2, Operational Guidance and Installation Procedures (AGD-IGS.1) is also available on this site.

1.6.1.6 TOE Delivery

The TOE and accompanying documentation are delivered from the OpenText software download site:

<https://sld.microfocus.com/mysoftware/download/downloadCenter>

The TOE is delivered as an iso file `secure-data-appliance-7.0.2-100050.iso`



Software Downloads

Account Name: laurie.odellus@microfocus.com

Product: Voltage SecureData

Product Name: Voltage SecureData Appliance

Version: 7.0.2

Export Media Report Reset

Download Selected ☐ Show Superseded Patches Get Licenses

By downloading the software below, you agree, on behalf of or as the licensee of such software, that you have read and hereby accept that, unless otherwise agreed in writing by OpenText, the End User License Agreement and any associated Additional License Authorizations located [here](#), for such software shall apply and supersede different license terms that may be embedded within such software.

1.6.2 Logical Boundary

TSF	DESCRIPTION
Cryptographic Support	<p>The TOE is a product whose main function is to provide cryptographic algorithms to calling APIs. The TOE provides implementations of the following cryptographic capabilities:</p> <ul style="list-style-type: none"> Format Preserving Encryption (FPE); embedded Format Preserving Encryption (eFPE); Identity-Based Encryption (IBE); and Identity-Based Symmetric Encryption (IBSE) <p>The TOE generates master keys to derive encryption keys. These are destroyed when they are no longer required.</p>
Security Management	<p>The TOE provides administrators with the capabilities to configure, monitor and manage the TOE to fulfill the Security Objectives. Security Management principles relate to management of access control policies as well as management of events and incidents. Administrators configure the TOE with the Console via Web-based connection. The TOE provides an inactivity timeout mechanism.</p>
User Data Protection	<p>The TOE implements an access control SFP named <i>SecureData Access Control SFP</i>. This SFP determines and enforces the privileges associated with user roles. TOE also provides mechanism to protect data residing operational environment and called by TOE.</p>

TSF	DESCRIPTION
Security Audit	The TOE generates reports on the event analysis activities. Additionally, the TOE supports the provision of log data from each system component, such as user login/logout and incident/ticket management actions. It also records security events such as failed login attempts, etc. Audit trails can be stored for later review and analysis. Audit data is also collected by the TOE from the various devices that send event data, and the TOE analyzes this information against a set of correlation rules and filters.
Identification and Authentication	The TOE enforces individual I&A in conjunction with group/role based I&A mechanisms. Operators must successfully authenticate using a unique identifier and password prior to performing any actions on the TOE.
TOE Access	Inactive sessions are terminated by the TOE. If there is no activity for 15 minutes, the session is terminated. The user can also terminate their own session.
Trusted Path	The TOE provides a trusted channel, TLS 1.2, for communications with the Secure Data clients.

Table 3– Logical Boundary

1.6.2.1 TOE Security Functional Policies

The TOE supports the following Security Functional Policy:

- **SecureData Access Control SFP**

The TOE implements an access control SFP named *SecureData Access Control SFP*. This SFP determines and enforces the privileges associated with user roles.

2 Conformance Claims

2.1 CC Conformance Claim

The TOE is conformant to Common Criteria Version 3.1 CC Revision 5, April 2017 Part 2 conformant and Part 3 conformant.

2.2 PP Claim

The TOE does not claim conformance to any registered Protection Profile.

2.3 Package Claim

The TOE claims conformance to the EAL3 assurance package defined in Part 3 of the Common Criteria Version 3.1 Revision 5 (April 2017)). The TOE does not claim conformance to any functional package. The TOE EAL3 assurance package is augmented with ALC_FLR.3.

2.4 Conformance Rationale

No conformance rationale is necessary for this evaluation since this Security Target does not claim conformance to a Protection Profile.

3 Security Problem Definition

In order to clarify the nature of the security problem that the TOE is intended to solve, this section describes the following:

- Any known or assumed threats to the assets against which specific protection within the TOE or its environment is required.
- Any organizational security policy statements or rules with which the TOE must comply
- Any assumptions about the security aspects of the environment and/or of the manner in which the TOE is intended to be used.

This chapter identifies assumptions as *A.assumption*, threats as *T.threat* and policies as *P.policy*.

3.1 Threats

The following are threats identified for the TOE and the IT System the TOE monitors. The TOE itself has threats and the TOE is also responsible for addressing threats to the environment in which it resides. The assumed level of expertise of the attacker for all threats is unsophisticated.

THREAT	DESCRIPTION
T.NO_AUTH	An unauthorized user may gain access to the TOE, possibly through an unattended authorized user session to alter the TOE configuration. The asset is the configuration of the TOE.
T.PRIV_ESC	An authorized user of the TOE may modify TOE configuration and is able to log on to the TOE as a person with higher authorization than his/her assigned security privileges. The asset is the configuration of the TOE.
T.DISCLOSE	An attacker may attempt to disclose and determine meaningful information from data collected and/or analyzed by TOE by bypassing a security mechanism. The asset is the sensitive data such as credit card numbers and Social Security numbers.
T.BRUTE_FORCE	An unauthorized user may gain access to the TOE by guessing valid user credentials to gain access to the TOE. The asset is user credentials.
T.COMPROMISE	An unauthorized user may attempt to modify or destroy audit data thus removing evidence of unauthorized or malicious activity. The asset is audit data.
T.CRYPTO	An unauthorized person or unauthorized IT entity may be able to view, modify, delete security related information that is sent between trusted devices. The asset is TSF data and user data.

Table 4 – Threats on the TOE

3.2 Assumptions

This section describes the security aspects of the environment in which the TOE is intended to be used. The TOE is assured to provide effective security measures in a co-operative non-hostile environment only if it is installed, managed, and used correctly. The following specific conditions are assumed to exist in an environment where the TOE is employed.

ASSUMPTION	DESCRIPTION
A.MANAGE	There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.
A.PROTECT	The TOE software critical to security policy enforcement will be protected from unauthorized modification. The operational environment provides domains for the isolation of security functionality.
A.PHYSICAL	The environment for the TOE is in a secure facility. There is no unauthorized access to the TOE.
A.TIMESOURCE	The TOE has a trusted source for system time via NTP server.
A.LDAP	The LDAP sever is a trusted IT entity.

Table 5 – Assumptions

4 Security Objectives

4.1 Security Objectives for the TOE

OBJECTIVE	DESCRIPTION
O.AUDIT	The TOE shall keep audit records.
O.PASSWORD	The TOE shall enforce a password policy.
O.I_AND_A	The TOE requires that a user be identified and authenticated before given access to TOE functions.
O.TIMEOUT	The TOE shall enforce session timeouts.
O.SEC_ACCESS	The TOE shall ensure that only those authorized users and applications are granted access to security functions and associated data.
O.PROTECT	TOE shall provide such mechanism in order to protect itself against attempts by unauthorized users via bypassing or tampering TSF which leads to unauthorized access to its data or to deny access to legitimate users.
O.COMMS	The TOE shall protect the confidentiality and the integrity of data passed between itself and the trusted entity using cryptographic functions.

Table 6 – Security Objectives for the TOE

4.2 Security Objectives for the Operational Environment

OBJECTIVE	DESCRIPTION
OE.TIME	The TOE operating environment shall provide an accurate timestamp.
OE.ENV_PROTECT	The TOE operating environment shall provide mechanisms to isolate the TOE Security Functions (TSF).
OE.LDAP	The environment will provide an LDAP Server.
OE.PHYSICAL	The TOE is protected in a secure facility with no unauthorized access to the TOE.
OE.PERSONNEL	Authorized administrators are non-hostile and follow all administrator guidance and must ensure that the TOE is delivered, installed, managed, and operated in a manner that maintains the TOE security objectives. Any operator of the TOE must be trusted to not disclose their authentication credentials. Authorized administrators are also required to manage and administer the TOE in a secure manner. Authorized administrators are competent and security aware personnel in accordance with the administrator documentation.

Table 7 – Security Objectives for the Operational Environment

4.3 Security Objectives Rationale

OBJECTIVES THREATS	O.AUDIT	O.PASSWORD	O.I_AND_A	O.SEC_ACCESS	O.PROTECT	O.TIMEOUT	O.COMMS	OE.TIME	OE.ENV_PROTECT	OE.PERSONNEL
T.NO_AUTH			X	X		X		X		
T.PRIV_ESC			X	X						
T.DISCLOSE					X		X		X	
T.BRUTE_FORCE		X		X						
T.COMPROMISE	X			X						
T.CRYPTO				X			X			

Table 8 - Threats to Objectives Mapping

OBJECTIVES ASSUMPTIONS	OE.TIME	OE.ENV_PROTECT	OE.PHYSICAL	OE.PERSONNEL	OE.LDAP
A.MANAGE				X	
A.TIMESOURCE	X				
A.PROTECT		X			
A.PHYSICAL			X		
A.LDAP					X

Table 9 - Assumptions to Objectives Mapping

4.4 Rationale for TOE Threats and Assumptions

ASSUMPTION/ THREAT/ POLICY	RATIONALE
A.MANAGE	This assumption is addressed by <ul style="list-style-type: none"> OE.PERSONNEL, which ensures that the TOE is managed and administered in a secure manner by a competent and security aware personnel in accordance with the administrator documentation. This objective also ensures that those responsible for the TOE install, manage, and operate the TOE in a secure manner
A.PROTECT	This assumption is addressed by <ul style="list-style-type: none"> OE.ENV_PROTECT which ensures the environment has a secure domain for TOE execution.
A.PHYSICAL	This assumption is addressed by <ul style="list-style-type: none"> OE.PHYSICAL which ensures that the TOE is protected in a secure facility with no unauthorized access to the TOE.
A.TIMESOURCE	This assumption is addressed by <ul style="list-style-type: none"> OE.TIME which ensures the TOE receives a reliable timestamp.
A.LDAP	This assumption is addressed by <ul style="list-style-type: none"> OE.LDAP which ensures the environment provides an LDAP Server.
T.NO_AUTH	This threat is countered by the following: <ul style="list-style-type: none"> O.SEC_ACCESS, which ensures that the TOE allows access to the security functions, configuration, and associated data only by authorized users and applications. O.I_AND_A which ensures a user is identified and authenticated before they can access TOE functionality. O.TIMEOUT ensures that an unattended session times out. OE.TIME which ensures the TOE receives a reliable timestamp
T.PRIV_ESC	This threat is countered by <ul style="list-style-type: none"> O.SEC_ACCESS, which ensures that the TOE allows access to the security functions, configuration, and associated data only by authorized users and applications. O.I_AND_A which ensures a user is identified and authenticated before the can access TOE functionality.
T.DISCLOSE	This threat is countered by <ul style="list-style-type: none"> O.PROTECT which ensures the TOE provides self-protection. O.COMM, which protects data entering the TOE or being transferred between parts of the TOE. OE.ENV_PROTECT which ensures the environment has a secure domain for TOE execution
T.BRUTE_FORCE	This threat is countered by <ul style="list-style-type: none"> O.PASSWORD which ensures the TOE provides and manages a password policy. O.SEC_ACCESS, which ensures that the TOE allows access to the security functions, configuration, and associated data only by authorized users and applications.

ASSUMPTION/ THREAT/ POLICY	RATIONALE
T.COMPROMISE	This threat is countered by <ul style="list-style-type: none">• O.SEC_ACCESS, which ensures that the TOE allows access to the security functions, configuration, and associated data only by authorized users and applications.• O.AUDIT which ensures that events are recorded and associated with users causing those events.
T.CRYPTO	This threat is countered by <ul style="list-style-type: none">• O.COMMS which protects data entering the TOE or being transferred between parts of the TOE.• O.SEC_ACCESS which cryptographically protects TSF and user data from disclosure.

Table 3 – Mapping of Threats, Policies, and Assumptions to Objectives

5 Extended Components Definition

There are no extended components.

6 Security Requirements

The security requirements that are levied on the TOE and the IT environment are specified in this section of the ST.

6.1 Security Functional requirements

CLASS HEADING	CLASS_FAMILY	DESCRIPTION
Security Audit	FAU_GEN.1	Audit Data Generation
	FAU_GEN.2	User Identity Association
	FAU_SAR.1	Audit Review
	FAU_SAR.3	Selectable Audit Review
	FAU_STG.1	Protected Audit Trail Storage
User Data Protection	FDP_ACC.1	Subset Access Control
	FDP_ACF.1	Security Attribute Based Access Control
Cryptographic Support	FCS_CKM.1(1)	Cryptographic Key Generation
	FCS_CKM.1(2)	Cryptographic Key Generation
	FCS_CKM.4	Cryptographic Key Destruction
	FCS_COP.1(1)	Cryptographic Operation
	FCS_COP.1(2)	Cryptographic Operation
Identification and Authentication	FIA_ATD.1	User Attribute Definition
	FIA_SOS.1	Verification of Secrets
	FIA_UAU.2	User Authentication before Any Action
	FIA_UAU.5	Multiple Authentication Mechanisms
	FIA_UID.2	User Identification before Any Action
Security Management	FMT_MSA.1	Management of Security Functions
	FMT_MSA.3	Static Attribute Initialization
	FMT_SMF.1	Specification of Management Functions
	FMT_SMR.1	Security Roles
TOE Access	FTA_SSL.3	TSF-initiated termination
	FTA_SSL.4	FTA_SSL.4 – User-initiated termination
	FTA_TSE.1	TOE Session Establishment
Trusted Path/Channels	FTP_ITC.1	Inter-TSF Trusted Channel
	FTP_TRP.1	Trusted Path

Table 4 – Security Functional Requirements

6.1.1 Security Audit (FAU)

6.1.1.1 FAU_GEN.1 Audit Data Generation

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) [the following auditable events:
 - All use of the user identification mechanisms
 - All use of the user authentication mechanisms
 - Use of the management functions]

- 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, 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, [none].

6.1.1.2 *FAU_GEN.2 User Identity Association*

- FAU_GEN.2.1** For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

6.1.1.3 *FAU_SAR.1 Audit Review*

- FAU_SAR.1.1** The TSF shall provide [administrators] with the capability to read [all audit information] from the audit records.
- FAU_SAR.1.2** The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

6.1.1.4 *FAU_SAR.3 Selectable Audit Review*

- FAU_SAR.3.1** The TSF shall provide the ability to apply [search and selection] of audit data based on [the following criteria:
- Search based on specified search criteria
 - Selection based on time interval, specification of audit record fields and specific values of selected fields
-].

6.1.1.5 *FAU_STG.1 Protected Audit Trail Storage*

- FAU_STG.1.1** The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.
- FAU_STG.1.2** The TSF shall be able to *detect* unauthorised modifications to the stored audit records in the audit trail.

6.1.2 Cryptographic Support (FCS)

6.1.2.1 *FCS_CKM.1(1) Cryptographic Key Generation*

- FCS_CKM.1.1(1)** The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [Hash-based DRBG] and specified cryptographic key sizes [128, 192, 256 bits] that meet the following: [NIST SP 800-90A].

6.1.2.2 *FCS_CKM.1(2) Cryptographic Key Generation*

FCS_CKM.1.1(2) The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [IBE BF private key generation; IBE BB1 private key generation] and specified cryptographic key sizes [3072, 4096 bits] that meet the following: [RFC 5091: IdentityBased Cryptography Standard (IBCS) #1: Supersingular Curve Implementations of the BF and BB1 Cryptosystems].

6.1.2.3 *FCS_CKM.4 Cryptographic Key Destruction*

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

6.1.2.4 *FCS_COP.1(1) Cryptographic Operation*

FCS_COP.1.1(1) The TSF shall perform [symmetric encryption and decryption] in accordance with a specified cryptographic algorithm [AES in CBC mode; AES in FF1 mode; Identity-Based Symmetric Encryption (IBSE) using AES in CBC or EME* mode; AES GCM mode] and cryptographic key sizes [128, 192, 256 bits] that meet the following: [FIPS 197; NIST SP 800-38A (AES in CBC mode); NIST SP 800-38G (AES in FF1 mode); P1619.2 – IEEE Standard for Wide-Block Encryption for Shared Storage Media (AES in EME* mode; SP 800-38D Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC].

6.1.2.5 *FCS_COP.1(2) Cryptographic Operation*

FCS_COP.1.1(2) The TSF shall perform [asymmetric encryption and decryption] in accordance with a specified cryptographic algorithm [Boneh-Franklin Identity Based Encryption; Boneh-Boyer Identity Based Encryption] and cryptographic key sizes [3072,4096 bits] that meet the following: [1363.3-2013 - IEEE Standard for Identity-Based Cryptographic Techniques using Pairings].

6.1.3 User Data Protection (FDP)

6.1.3.1 *FDP_ACC.1 Subset Access Control*

FDP_ACC.1.1 The TSF shall enforce the [SecureData Access Control SFP] on [

- Subjects: (administrator, Auditor, business applications, web applications)
- Objects: Keys
- Operations: protect; access; mask access

].

6.1.3.2 FDP_ACF.1 Security attribute-based access control

FDP_ACF.1.1 The TSF shall enforce the [SecureData Access Control SFP] to objects based on the following: [

- Subjects: administrator, Auditor, Business applications, web applications
- Objects: Keys
- Operations: protect; access; mask access

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects as shown in the table below

Subjects	Administrator	Client with IP Authorization	Auditor	Client with Identity Authorization	
Access Level Operations	Full Access			No Access (Read only access)	Masked Access*
Configure the Identity matching criteria	X	-	Read only	-	-
Create/delete user accounts	X	X	Read only	-	-
Modify user accounts	X	X	Read only	-	-
Configure masked access settings	X	-	Read only	-	-
Decryption of the protected data	X	X	X	-	X
Encryption of the unprotected data	X	X	X	-	X
View the protected data	X	X	Read only	X	X

*This Access level allows a client program to function only allowed portions of the accessed data].

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [no explicit authorization rules].

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [no explicit denial rules].

6.1.4 Identification and Authentication (FIA)

6.1.4.1 *FIA_ATD.1 User attribute definition*

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: [

- User Identity
- Authentication Data].

6.1.4.2 *FIA_SOS.1 Verification of Secrets*

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet [the following constraints:

- Minimum length of 8 characters
- At least 1 numeric character
- At least 1 uppercase character
- At least 1 lowercase character].

6.1.4.3 *FIA_UAU.2 User Authentication Before Any Action*

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

6.1.4.4 *FIA_UAU.5 Multiple Authentication Mechanisms*

FIA_UAU.5.1 The TSF shall provide [Local Password, Remote LDAP Authentication, Shared Secret, Client Certificate] to support user authentication.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the [following rules:

- For administrators:
 - If login access via Remote LDAP Authentication mechanism is enabled, administrators can login using credentials managed by a configured LDAP resource
 - If login access via Remote LDAP Authentication mechanism is not enabled, or no configured LDAP resource is available, administrators authenticate using Local Password
- Clients authenticate using an Authentication Method configured for their district, which could be:
 - Shared Secret

- Remote LDAP Authentication
- Client Certificate.

For successful authentication, the Client identity must match the Identity Pattern configured for the Authentication Method and the Client IP address must match an IP Address configured for the Authentication Method.]

6.1.4.5 *FIA_UID.2 User identification before any action*

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

6.1.5 Security Management (FMT)

6.1.5.1 *FMT_MSA.1 Management of Security Functions*

FMT_MSA.1.1 The TSF shall enforce the [SecureData Access Control SFP] to restrict the ability to *change_default, query, modify, delete* [no other operations] the security attributes [user identity, authentication data] to [administrator].

6.1.5.2 *FMT_MSA.3 Static Attribute Initialisation*

FMT_MSA.3.1 The TSF shall enforce the [SecureData Access Control SFP] to provide [restrictive [no other property]] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the [administrator] to specify alternative initial values to override the default values when an object or information is created.

6.1.5.3 *FMT_SMF.1 Specification of Management Function*

FMT_SMF.1.1 The TSF shall be capable of performing the following **security** management functions: [

- Manage Key Management policy
- Manage authentication methods
- Manage Identity Authorization and IP Authorization rules
- Manage mask settings
- Manage date and time
- Manage user accounts
- Manage LDAP resources
- Manage network access

].

6.1.5.4 FMT_SMR.1 Security Roles

FMT_SMR.1.1 The TSF shall maintain the roles: [web service clients, business clients, administrator, Auditor].

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

6.1.6 TOE ACCESS (FTA)

6.1.6.1 FTA_SSL.3 TSF-Initiated Termination

FTA_SSL.3.1 The TSF shall terminate an interactive session after a [time interval of user inactivity of 15 minutes].

6.1.6.2 FTA_SSL.4 User Initiated Termination

FTA_SSL.4.1 The TSF shall allow user-initiated termination of the user's own interactive session.

6.1.6.3 FTA_TSE.1 TOE Session Establishment

FTA_TSE.1.1 The TSF shall be able to deny session establishment based on [IP Address].

6.1.7 Trusted Path/Channels (FTP)

6.1.7.1 FTP_ITC.1 Inter-TSF trusted channel

FTP_ITC.1.1 The TSF shall provide a communication path between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the communicated data from modification or disclosure.

FTP_ITC.1.2 The TSF shall permit *the TSF* to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for [LDAP authentication requests].

6.1.7.2 FTP_TRP.1 Trusted Path

FTP_TRP.1.1 The TSF shall provide a communication path between itself and *remote* users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from *modification, disclosure*.

FTP_TRP.1.2 The TSF shall permit *remote users* to initiate communication via the trusted path.

FTP_TRP.1.3 The TSF shall require the use of the trusted path for *initial user authentication*, [all remote administrative actions].

6.2 Security Assurance Requirements Rationale

The ST specifies Evaluation Assurance Level 3, augmented with ALC_FLR.3. EAL3+ was chosen because it is based upon good commercial development practices with thorough functional testing. EAL3+ provides the developers and users a moderate level of independently assured security in conventional commercial TOEs. The threat of malicious attacks is not greater than low, the security environment provides physical protection, and the TOE itself offers a very limited interface, offering essentially no opportunity for an attacker to subvert the security policies without physical access.

Requirement Class	Component
ADV: Development	ADV_ARC.1 – Security architecture description
	ADV_FSP.3 – Functional specification with complete summary
	ADV_TDS.2 – Architectural Design
AGD: Guidance documents	AGD_OPE.1 – Operational user guidance
	AGD_PRE.1 – Preparative procedures
ALC: Life-cycle support	ALC_CMC.3 – Use of a CM system
	ALC_CMS.3 – Parts of the TOE CM coverage
	ALC_DEL.1 – Delivery procedures
	ALC_DVS.1 - Identification of security measures
	ALC_FLR.3 – Systematic Flaw Remediation
	ALC_LCD.1 - Developer Defined Life-Cycle Model
ASE: Security Target evaluation	ASE_CCL.1 – Conformance claims
	ASE_ECD.1 – Extended components definition
	ASE_INT.1 – ST introduction
	ASE_OBJ.2 – Security objectives
	ASE_REQ.2 – Derived security requirements
	ASE_SPD.1 – Security problem definition
	ASE_TSS.1 – TOE summary specification

Requirement Class	Component
ATE: Tests	ATE_COV.2 – Evidence of coverage
	ATE_DPT.1 – Testing: Basic Design
	ATE_FUN.1 – Functional testing
	ATE_IND.2 – Independent testing – sample
AVA: Vulnerability assessment	AVA_VAN.2 – Vulnerability analysis

Table 5 – Assurance Measures

6.3 Dependency Rationale

SFR CLAIM	DEPENDENCIES	HOW MET
FAU_GEN.1	FPT_STM.1	Provided by the NTP Server in the environment
FAU_GEN.2	FAU_GEN.1 FIA_UID.1	Included Covered by FIA_UID.2
FAU_SAR.1	FAU_GEN.1	Included
FAU_SAR.3	FAU_SAR.1	Included
FAU_STG.1	FAU_GEN.1	Included
FCS_CKM.1(1)	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_COP.1(1) and FCS_COP.1(2) are included. FCS_CKM.4 is included.
FCS_CKM.1(2)	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_COP.1(1) and FCS_COP.1(2) are included. FCS_CKM.4 is included.
FCS_CKM.4	[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.1(1) and FCS_CKM.1(2) are included

SFR CLAIM	DEPENDENCIES	HOW MET
FCS_COP.1(1)	[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.1(1) and FCS_CKM.4 is included
FCS_COP.1(2)	[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.1(2) are included FCS_CKM.4 is included
FDP_ACC.1	FDP_ACF.1	Included
FDP_ACF.1	FDP_ACC.1 FMT_MSA.3	Included Included
FIA_ATD.1	none	N/A
FIA_SOS.1	none	N/A
FIA_UAU.2	FIA_UID.1	FIA_UID.2
FIA_UAU.5	none	N/A
FIA_UID.2	none	N/A
FMT_MSA.1	FDP_ACC.1 FMT_SMF.1 FMT_SMR.1	Included Included Included
FMT_MSA.3	FMT_MSA.1 FMT_SMR.1	Included Included
FMT_SMF.1	none	N/A
FMT_SMR.1	FIA_UID.1	Covered by FIA_UID.2
FTA_SSL.3	none	N/A
FTA_SSL.4	none	N/A
FTA_TSE.1	none	N/A
FTP_ITC.1	none	N/A
FTP_TRP.1	none	N/A

Table 13 – Dependency Rationale

6.4 Objectives to SFR Mapping

SFR \ OBJECTIVE							
	O.AUDIT	O.PASSWORD	O.I_AND_A	O.SEC_ACCESS	O.COMMS	O.TIMEOUT	O.PROTECT
FAU_GEN.1	✓						
FAU_GEN.2	✓						
FAU_SAR.1	✓						
FAU_STG.1	✓						
FCS_CKM.1(1)							✓
FCS_CKM.1(2)							✓
FCS_CKM.4							✓
FCS_COP.1(1)							✓
FCS_COP.1(2)							✓
FDP_ACC.1				✓			
FDP_ACF.1				✓			
FIA_ATD.1				✓			
FIA_SOS.1		✓					
FIA_UAU.2			✓				
FIA_UAU.5			✓				
FIA_UID.2			✓				
FMT_MSA.1				✓			
FMT_MSA.3				✓			
FMT_SMF.1				✓			
FMT_SMR.1				✓			
FTA_SSL.3				✓		✓	
FTA_SSL.4				✓		✓	
FTA_TSE.1				✓			
FTP_ITC.1					✓		
FTP_TRP.1					✓		

Table 14 – Objectives to SFR Mapping

6.5 Objectives to SFRs Rationale

Objective	RATIONALE
O.AUDIT	<p>This objective ensures that events are audited.</p> <ul style="list-style-type: none"> FAU_GEN.1 requires that events be audited. FAU_GEN.2 requires that audited events be associated with a user. FAU_SAR.1 requires that audit records are readable and that only administrators can view audit records. FAU_SAR.3 requires that audit records are searchable. FAU_STG.1 requires the TOE to detect modifications to the audit records.
O.PASSWORD	<p>This objective requires the TOE to enforce a password policy.</p> <ul style="list-style-type: none"> FIA_SOS.1 requires that the administrator sets a password policy that is enforced by the TOE.
O.I_AND_A	<p>This objective requires all users of the TOE to be identified and authenticated.</p> <ul style="list-style-type: none"> FIA_UAU.2 requires the TOE to enforce authentication of all users prior to accessing the TOE. FIA_UID.2 requires the TOE to enforce identification of all users prior to accessing the TOE. FIA_UAU.5 requires one of the methods listed must be used for authentication of the user.
O.SEC_ACCESS	<p>This objective ensures that the TOE allows access to the security functions, configuration, and associated data only by authorized users and applications.</p> <ul style="list-style-type: none"> FDP_ACC.1 requires that all user actions resulting in the access to TOE security functions and configuration data are controlled. FDP_ACF.1 supports FDP_ACC.1 by ensuring that access to TOE security functions, configuration data, audit logs, and account attribute. FIA_ATD.1 specifies security attributes for users of the TOE FMT_MSA.1 specifies that only privileged administrators can manage security attributes. FMT_MSA.3 ensures that all default values of security attributes are restrictive in nature as to enforce the access control policy for the TOE. The Administrator can specify alternative initial values that will override default values. FMT_SMF.1 and FMT_SMR.1 support the security functions relevant to the TOE and ensure the definition of an authorized administrator role. FTA_SSL.3 requires the TSF lock after a period of inactivity. FTA_SSL.4 requires the user be able to initiate a session termination. FTA_TSE.1 The TOE is able to deny session establishment based on IP addresses.

Objective	RATIONALE
O.PROTECT	<p>This objective ensures that the TOE provides such mechanism in order to protect sensitive data</p> <ul style="list-style-type: none"> FCS_CKM.1(1) and FCS_CKM.1(2) ensure that the correct keys are generated for cryptographic use. FCS_CKM.4 ensures keys are destroyed if no longer required. FCS_COP.1(1) and FCS_COP.1(2) ensures generated keys are used to protect data.
O.COMMS	<p>This objective requires that any communications with or within the TOE are protected.</p> <ul style="list-style-type: none"> FPT_ITC.1 provides a trusted channel between the TOE and trusted IT products. FTP_TRP.1 provides a trusted communications path with the TOE.
O.TIMEOUT	<p>This objective ensures that TOE access is enforced.</p> <ul style="list-style-type: none"> FTA_SSL.3 requires the TSF lock after a period of inactivity. FTA_SSL.4 requires the user be able to initiate a session termination.

Table 15 – Objectives to SFR Rationale

7 TOE Summary Specification

This section presents the Security Functions implemented by the TOE.

7.1 TOE Security Functions

The security functions performed by the TOE are as follows:

- Security Audit
- Cryptographic Support
- User Data Protection
- Identification and Authentication
- Security Management
- TOE Access
- Trusted Path

7.2 Security Audit

TOE is designed to generate audit records (events) of security relevant and other events as they occur.

The events that can cause an audit record to be generated include:

- Starting and stopping the audit function
- All use of the user identification mechanism
- All use of the user authentication mechanism
- Use of the management functions
- Termination of an interactive session by the user.

Generated audit records include the following information: date and time of the event; type of event; subject identity; and a description of the event and its outcome. Generated audit events resulting from the actions of identified users include the identity of the user that caused the event.

The TOE provides administrators with capabilities to read all of the audit information included in the generated audit events, using the **Events** tab of the Management Console. This provides access to the **Event Viewer**, which enables events to be displayed in a set of standard reports. The Event Reports are organized into the following categories:

- Recent Events—reports showing all recent events and events by category.
- Key Management—reports with events related to key requests, failed authentications, and key denials.
- Web Service—reports with events related to failed authentications and failed authorizations from Web Services interfaces.
- Console—reports with events related to console logins, shell logins that include both administrator and root logins, administrator account activity, and all console events.

All reports are displayed using the same basic layout and presenting the same types of information. Displayed reports have the following sections:

- Search field—allows the administrator to specify search criteria using the Event Viewer search commands. Searches return only the events that meet the search criteria.
- Time Interval—drop-down list that enables the administrator to select the time interval for displayed events. The default is all time.
- Graph Area—shows in a graphical format the number of events of the selected type over the specified time interval.
- Selected Fields—allows the administrator to select specific data fields to display in the third line of each event in the report. The administrator can view a list of available fields and can select the order in which the selected fields are presented on the display.
- Events—displays the specific data associated with each reported event. By default, events are displayed in a list.

Each event includes three lines. The first line displays either a green check with the word “Valid” (indicating the event is intact in the Event Viewer database) or a red exclamation mark with the word “Tampered” (indicating the event has been altered in the Event Viewer database). The second line displays the complete content of the audit record, while the third line displays a subset of the information from the second line, including just the information from the fields selected in the **Selected Fields** section of the display. The administrator can click any of the data in the first or second lines to filter the report so that it shows only events that also include that particular data value.

The Security Audit function is designed to satisfy the following security functional requirements:

- FAU_GEN.1
- FAU_GEN.2
- FAU_SAR.1
- FAU_SAR.3
- FAU_STG.1

7.3 Cryptographic Support

The TOE provides implementations of the following cryptographic capabilities:

Cryptographic keys are generated and used and when no longer required, are destroyed.

FCS_CKM.1(1)

FCS_CKM.1(2)

FCS_CKM.4

FCS_COP.1(1)

FCS_COP.1(2)

7.4 User Data Protection

The TOE is able to restrict the operations available to clients using the SecureData Web Service APIs.

Web service client authorization is based on authentication controls (specified in Identity Authorization rules) and IP address (specified in IP Authorization rules). The administrator configures Identity Authorization and IP Authorization rules on the **Web Service** tab of the Management Console and the client program is allowed to perform a requested operation only if both the Identity Authorization rules and the IP Authorization rules permit the operation.

7.4.1 Identity Authorization

Identity authorization is granted based on two pieces of information passed in by the client—the identity string and the authentication credentials, which can be a shared secret, <username>:<password>, or client certificate.

An Identity Authorization rule specifies the following:

- One or more identity patterns against which client identities are compared
- Criteria for matching against the client authentication credentials. The following matching criteria can be specified:
 - Shared Secret—an agreed-upon value configured by the administrator that the client application needs to know
 - User Name Patterns—regular expressions of user names for which the rule is applicable
 - LDAP Group Lookup—specifies a lookup in a configured LDAP resource and specified LDAP groups.

The matching behavior depends on both the matching criteria specified in the rule and the authentication mechanism used by the client, as follows:

- If the client uses Shared Secret authentication, the shared secret passed in the `authInfo` parameter is matched with the shared secret specified in the rule. A value of “*” in the rule matches any valid shared secret.
- If the client uses Username and Password authentication and User Name Pattern is specified, the username passed in the `authInfo` parameter is matched with the user name patterns specified in the rule.
- If the client uses Username and Password authentication and LDAP Group Lookup is specified, the Web Service tests if the username passed in the `authInfo` parameter is a member of an LDAP group specified in the rule.
- If the client uses Certificate authentication and User Name Pattern is specified, the Web Service tests if the `CN`, `Subject Alternate Name`, or `Subject` attributes in the certificate match the user name patterns specified in the rule.

- If the client uses Certificate authentication and LDAP Group Lookup is specified, the Web Service tests if the `CN` or `Subject Alternate Name` attributes in the certificate are members of the groups specified in the rule.
- The operations permitted to the client if the client's identity and authentication credentials match the rule. The following operations can be specified:
 - Protect—allows an authorized client to protect data using the Protect methods available in the Web Services. If Protect is not specified, the rule will allow only decryption operations to be performed
 - Access Level—specifies the level of access granted to the client when decrypting data. The following levels are defined:
 - No Access—prevents clients matching the identity pattern and client authentication criteria from accessing any protected data
 - Masked Access—allows clients matching the identity pattern and client authentication criteria to access data, but returns only a subset of the accessed data and masks the remainder of the data
 - Full Access—allows clients matching the identity pattern and client authentication criteria to access and view all of the data.

The TOE performs Identity Authorization rule matching on the client authentication match criterion first; if that passes, the validity of the identity is verified. For example, if a rule's authentication match criterion is Shared Secret, the Authorization method used in the web services call must also be Shared Secret, and the rule is applicable only if the shared secret passed in the web service call matches the value specified in the rule. If the shared secret passed in matches, but the identity does not match an identity pattern specified in the rule, the rule is not applied.

All configured Identity Authorization rules are evaluated independently. Therefore, if any rule matches, authorization is granted and the operation can be performed. Furthermore, if multiple rules match, the rule granting the highest authorization applies. For example, if one rule allows full access and another rule allows no access, the full access takes precedence.

The TOE specifies default Identity Authorization rules for Shared Secret and User Name Patterns that grant the Protect operation and Full Access.

7.4.2 IP Authorization

IP authorization is granted based on the client's IP address. If a client IP address does not match any of the IP address patterns specified in the IP authorization rules, the client program is not able to perform any of the Web Service operations.

An IP Authorization rule specifies the following:

- One or more IP address patterns against which client IP addresses are compared
- The operations permitted to the client if the client's IP address matches the rule. The following operations can be specified:

- Protect—allows an authorized client to protect data using the Protect methods available in the Web Services. If Protect is not specified, the rule will allow only decryption operations to be performed
- Access Level—specifies the level of access granted to the client when decrypting data. The following levels are defined:
 - No Access—prevents clients matching the identity pattern and client authentication criteria from accessing any protected data
 - Masked Access—allows clients matching the identity pattern and client authentication criteria to access data, but returns only a subset of the accessed data and masks the remainder of the data
 - Full Access—allows clients matching the identity pattern and client authentication criteria to access and view all of the data.

The TOE specifies a default IP Authorization rule that grants the Protect operation and Full Access to all clients from any IP address.

The User Data Protection security function satisfies the following security functional requirements:

- FDP_ACC.1, FDP_ACF.1—the TOE enforces an access control policy on Web Service clients that authorizes clients to perform protection and access operations on data based on the client's identity, authentication credentials, and IP address.

7.5 Identification and Authorization

The TOE distinguishes between two types of users—administrators, who configure and manage the TOE, and clients, who request key management services of the TOE via a supported SecureData software client. The Identification and Authentication security function provides the capability for the TOE to identify and authenticate both administrators and clients.

7.5.1 Administrator I&A

The TOE requires administrators to be successfully identified and authenticated before they can access any of the management functions provided by the TOE. The TOE offers both a locally connected console and a network accessible interface over HTTPS (the Management Console) for interactive administrator sessions. There is a password policy that is applied only to local users.

The TOE supports the local (i.e., on device) definition of administrators with usernames and passwords. Additionally, the TOE can be configured to use an LDAP directory to support remote user authentication, enabling LDAP login access to the Management Console. Once configured, any user belonging to the specified LDAP group(s) is granted automatic login access to the Management Console without having to explicitly create a local account for that user.

After configuring LDAP Access to the Management Console, users belonging to the configured LDAP group(s) can log into the Management Console using their LDAP username and password. The system automatically verifies the login credentials against the configured LDAP server, and then verifies that the user belongs to at least one of the configured groups (either directly or through a group hierarchy). If both checks succeed, the user is logged into the console, and a new user account entry is automatically created for that user in the console database and is listed in the **Users** table displayed by the Management Console under the **Administration** tab.

If the LDAP check fails, the TOE checks the user's credentials against its list of local users and logs the user in if it finds a match. This allows the administrator to maintain non-LDAP user accounts that can be used to login to the Management Console even if the LDAP server is down.

7.5.2 Client I&A

When the TOE receives requests for encryption or decryption keys from SecureData applications running on client machines, the clients must first be authenticated. The administrator configures one or more authentication methods for a district. An authentication method defines rules for authenticating the identity of a key requester, including identity patterns, IP addresses, and the type of authentication. The TOE supports the following client authentication types in its evaluated configuration:

- Shared secret—this authentication type uses a configured shared secret to authenticate with the SDA
- LDAP Username and Password—this authentication type uses a username and password pair to authenticate with the SDA using LDAP
- Client Certificate—this authentication type uses a client certificate to authenticate with the Key Server.

For successful authentication, the client identity must also match the identity pattern configured for the authentication method and the client IP address must match an IP address configured for the authentication method.

The Identification and Authentication function satisfies the following security functional requirements:

- FIA_ATD.1—the TOE maintains the following security attributes associated with each administrator: user identity; and authentication data.
- FIA_SOS.1—the TOE enforces a password policy that ensures all secrets (i.e., passwords) associated with user accounts meet policy requirements.
- FIA_UAU.2—the TOE requires each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.
- FIA_UAU.5 – The TOE requires that one of the methods of authentication described be used.
- FIA_UID.2—the TOE requires each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

7.6 Security Management

When the SDA is installed, two default accounts are created—**admin** and **root**. The **admin** account is used to perform initial configuration using the Appliance Menu and administration functions using the Management Console, including creation of additional user accounts. The guidance documentation cautions not to log in as **root** unless following instructions in the guidance or troubleshooting with the help of technical support.

All users with an account on the SDA are considered administrators—each user has full access to all administrative functionality, except where the user account is disabled or restricted by dual control settings. Administrators use the Appliance Menu for initial configuration of the SDA and the Management Console for on-going management and administration.

The Appliance Menu is a simple hierarchical menu interface that enables the administrator to perform initial configuration tasks, such as configuring network settings, configuring date and time, and changing default passwords for the **admin** and **root** accounts.

The Management Console provides a web-based interface that provides the following management capabilities:

- Define the Key Management policy, which defines attributes of the Key Management Server
- Define attributes for Web Service API access, including authorizations and required authentication information
- Define the type of authentication that must be used by anyone requesting keys from the Key Management Server
- Define formats for all data types including credit cards, US Social Security numbers, regular numbers, dates, and variable-length and specified-format strings
- Manage mask settings for access using the SecureData Web Service APIs
- Control the type of access required for the network and specific configuration actions
- Manage the date and time
- Manage user accounts
- Manage LDAP resources.

The Security Management function satisfies the following security functional requirements:

- FMT_SMF.1—the TOE provides the capabilities necessary to manage the security of the TOE.
- FMT_SMR.1—the TOE defines a single role (administrator) that is assigned to every user with an account on the TOE.
- FMT_MSA.1, FMT_MSA.3

7.7 TOE Access

Users logged in to the Management Console are automatically logged out after 15 minutes of inactivity. Users are also able to terminate their interactive sessions with the Management Console by clicking **Logout** on the **Home** page.

The TOE can be configured to limit remote access to the Management Console to IP addresses matching administrator-configured IP addresses or address patterns. The administrator manages IP patterns via the **Network Access** page under the **Administration** tab of the Management Console. The list of addresses/address patterns can include overlapping and identical patterns and patterns can include “*” (matches any string of zero or more characters) and “?” (matches any single character) wildcards.

The TOE Access security function satisfies the following security functional requirements:

- FTA_SSL.3—the TOE terminates an interactive session after a time interval of user inactivity of 15 minutes.
- FTA_SSL.4—the TOE allows user-initiated termination of the user’s own interactive session.
- FTA_TSE.1—the TOE is able to deny session establishment based on the IP address of the source of a session request.

7.8 Trusted Path/Channels

The SDA provides a trusted channel to communicate securely with clients and with configured LDAP resources. The trusted channel is implemented using TLS. The use of TLS ensures all communication over the trusted channel is protected from disclosure and modification. The TOE will initiate the trusted channel when submitting an authentication request to a configured LDAP resource.

The SDA provides a trusted path for administrators to communicate with the SDA. The trusted path is implemented using HTTPS (i.e., TLS over HTTP) for access to the Management Console. Administrators initiate the trusted path by establishing an HTTPS connection (using a supported web browser) to the Management Console. The trusted path is used for initial authentication and all subsequent administrative actions. The use of HTTPS ensures all communication over the trusted path is protected from disclosure and modification.

In addition to the Management Console Server, which is the primary mechanism for administration, the SDA provides the Appliance Menu, which is used for initial configuration of the SDA after it has been installed. The administrator accesses the Appliance Menu by logging on to the SDA **admin** account using the SDA server's console.

The Trusted Path/Channels security function satisfies the following security functional requirements:

- FTP_ITC.1—the TOE provides a trusted channel for the TOE to communicate with SecureData clients and LDAP resources.
- FTP_TRP.1—the TOE provides a trusted path for administrators to communicate with the TOE, using HTTPS to access the Management Console.