Voltage
SecureData Appliance and SecureData Simple API
Security Target

Version 1.0
1 November 2017

Prepared for:

1140 Enterprise Way
Sunnyvale, CA 94089

Prepared By:

Accredited Testing and Evaluation Labs
6841 Benjamin Franklin Drive
Columbia, MD 21046

TABLE OF CONTENTS

LIST OF TABLES

1. Introduction

This section introduces the Target of Evaluation (TOE) and provides the Security Target (ST) and TOE identification, ST and TOE conformance claims, ST conventions, glossary and list of abbreviations.

The TOE is Micro Focus – Voltage SecureData Appliance v6.4 (SDA) with SecureData Simple API v5.10. 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 SecureData Simple API provides a set of functions that are callable from existing C, C#/.NET, and Java applications. It allows data protection functionality to be included into any such application and enables applications to communicate with the SDA to obtain keys.

The ST contains the following additional sections:

- TOE Description (Section 2)—provides an overview of the TOE and describes the physical and logical boundaries of the TOE
- Security Problem Definition (Section 3)—describes the threats and assumptions that define the security problem to be addressed by the TOE and its environment
- Security Objectives (Section 4)—describes the security objectives for the TOE and its operational environment necessary to counter the threats and satisfy the assumptions that define the security problem
- IT Security Requirements (Section 5)—specifies the security functional requirements (SFRs) and security assurance requirements (SARs) to be met by the TOE
- TOE Summary Specification (Section 6)—describes the security functions of the TOE and how they satisfy the SFRs
- Rationale (Section 7)—provides mappings and rationale for the security problem definition, security objectives, security requirements, and security functions to justify their completeness, consistency, and suitability.

1.1 Security Target, TOE and CC Identification

ST Title – Voltage SecureData Appliance and SecureData Simple API Security Target

ST Version – Version 1.0

ST Date – 1 November 2017

TOE Identification – SecureData Appliance v6.4 and SecureData Simple API 5.10

TOE Developer – Micro Focus – Voltage

Evaluation Sponsor – Micro Focus – Voltage

CC Identification – Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 4, September 2012

1.2 Conformance Claims

This ST and the TOE it describes are conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Components, Version 3.1 Revision 4, September 2012.
    - Part 2 Conformant
- Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Components, Version 3.1 Revision 4, September 2012.

- Part 3 Conformant

This ST and the TOE it describes are conformant to the following package:

- EAL2 Augmented (ALC_FLR.1)

1.3 Conventions

The following conventions are used in this document:

- Security Functional Requirements—Part 1 of the CC defines the approved set of operations that may be applied to functional requirements: iteration; assignment; selection; and refinement.
  - Iteration—allows a component to be used more than once with varying operations. In this ST, iteration is identified with a number in parentheses following the base component identifier. For example, iterations of FCS_COP.1 are identified in a manner similar to FCS_COP.1(1) (for the component) and FCS_COP.1.1(1) (for the elements).
  - Assignment—allows the specification of an identified parameter. Assignments are indicated using bold text and are enclosed by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
  - Selection—allows the specification of one or more elements from a list. Selections are indicated using bold italics and are enclosed by brackets (e.g., [***selection***]).
  - Refinement—allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").
- Other sections of the ST—other sections of the ST use bolding and/or different fonts (such as `Courier`) to highlight text of special interest, such as captions, commands, or filenames specific to the TOE.

1.4 Glossary

This ST uses a number of terms that have a specific meaning within the context of the ST and the TOE. This glossary provides a list of those terms and how they are to be understood within this ST.

Apache Hadoop	An open-source software framework used for distributed storage and processing of very large data sets.
district	Entity that provides access to a set of values that define how data can be protected, as well as to information about whether a key can be issued for a particular operation. The district domain name is a valid domain name that maps to the district name in the key generator and public parameter configuration files.
identity	A formatted string linking a client application to a cryptographic key managed by the TOE.
PKCS7	Public Key Cryptography Standard #7—the Cryptographic Message Syntax Standard, used to sign and/or encrypt messages under a PKI. It is defined in RFC 2315.
tokenization	A capability supported by the TOE that allows data in an application to be replaced by an alias or "token".
z/OS	An IBM mainframe operating system.

1.5 Abbreviations and Acronyms

The following abbreviations and acronyms are used in this ST:

AES	Advanced Encryption Standard
API	Application Programming Interface
CBC	Cipher Block Chaining—a mode of operation of AES

CC	Common Criteria
CSV	Comma Separated Values—a format for storing data in plain text in fields separated by commas
DNS	Domain Name System
DoS	Denial of Service
DRBG	Deterministic Random Bit Generator
EAL	Evaluation Assurance Level
ECB	Electronic Code Book—a mode of operation of AES
eFPE	embedded Format Preserving Encryption—an encryption method that encrypts data so that identity information is embedded into the ciphertext, preserving the length but not the character set of the input data
EME* (EME2)	ECB-Mix-ECB—a mode of operation of AES
FPE	Format Preserving Encryption—an encryption method that encrypts data so that the ciphertext has the same length and character set as the input data
GUI	Graphical User Interface
HSM	Hardware Security Module—a physical computing device that manages and safeguards cryptographic keys and may support other cryptographic operations
IBE	Identity-Based Encryption—an asymmetric encryption algorithm that encrypts data without preserving its length or character set
IBE BB1	Identity Based Encryption Boneh-Boyen—an identity-based encryption system proposed by Dan Boneh and Xavier Boyen in 2004
IBE BF	Identity Based Encryption Boneh-Franklin—an identity-based encryption system proposed by Dan Boneh and Matthew K. Franklin in 2001
IBSE	Identity-Based Symmetric Encryption—a non-format-preserving symmetric encryption algorithm
KDF	Key Derivation Function—a function that derives one or more cryptographic keys from a secret value such as a master key or password.
IT	Information Technology
LDAP	Lightweight Directory Access Protocol
NTP	Network Time Protocol
PIE	Page Integrated Encryption—a technology that encrypts sensitive user data in a web browser and allows that data to travel encrypted through intermediate application tiers
PKI	Public Key Infrastructure
RADIUS	Remote Authentication Dial-In User Service
REST	Representational state transfer—a software architecture for distributed systems, including RESTful API web services
RFC	Request for Comments
SAR	Security Assurance Requirement
SDA	SecureData Appliance
SFP	Security Function Policy
SFR	Security Functional Requirement
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SOAP	Simple Object Access Protocol—a protocol specification for exchanging structured information in the implementation of web services in computer networks.
SSN	Social Security Number
ST	Security Target
TCP	Transmission Control Protocol
TDE	Transparent Data Encryption—a technology employed by various vendors to encrypt database files.

TOE	Target of Evaluation
TSF	TOE Security Function
UDF	User Defined Function
UDP	User Datagram Protocol
WSDL	Web Services Description Language

2. TOE Description

The TOE, Micro Focus – Voltage SecureData Appliance, v6.4 (SDA), with SecureData Simple API v5.10, is a software product that 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 SecureData 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.

2.1 Overview

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, an API call, or a command line 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) in the operational environment.

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:

- Voltage SecureData Web—supports data protection within the browser of a merchant's customer.
- Voltage SecureData Mobile—supports data protection within a merchant's mobile app.
- Voltage SecureFile Command Line (CL)—supports encryption of entire files.

- Voltage SecureData Command Line (CL)—supports encryption of data within a CSV or Copybook file, or within a database table.
- Voltage SecureData Tokenization Command Line—supports tokenization of data within a CSV file.
- Voltage SecureData File Processor—supports protection of data within a CSV file or within a text file with fixed-width columns.
- Voltage SecureData for Teradata—provides user-defined functions (UDFs) that protect and access data stored in a Teradata database.
- Voltage SecureData Payments Software—supports credit card payment systems.
- z/OS-based Solutions—provide encryption software for z/OS systems.
- Voltage SecureData for Hadoop—includes both full-disk encryption and the ability to protect and access protected data in a Hadoop environment, as well as key management for Hadoop Transparent Data Encryption (TDE) requests.

Note, the SDA provides a FIPS mode of operation, but this cannot be used in the evaluated configuration, as it limits client access to only the REST API—in particular, the SecureData Simple API is not supported on a FIPS-enabled SDA.

2.2 TOE Components

The TOE is a software product that consists of the following components:

- Key Management Server
- Management Console
- Web Service Server (SOAP and REST APIs)
- SecureData Simple API.

Note, 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. As noted in Section 2.1 above, 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). As noted in Section 2.1 above, Voltage SecureData for Hadoop is not included within the scope of the evaluation.

2.2.1 Management Console

The Management Console provides a web-based interface to support centralized configuration and reporting across the Voltage 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 Voltage 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 Voltage SecureData deployment. Regardless of how the enterprise is using the Voltage SecureData data protection products, they must be configured through the Management Console.

2.2.2 Key Management Server

The Key Management Server supports centralized Voltage SecureData key management. Voltage 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.

2.2.3 Web Services Server

The Web Service Server provides a data protection interface that can be used by web applications capable of consuming 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 Voltage 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.

2.2.4 SecureData Simple API

The SecureData Simple API client provides a set of functions that are callable from existing C, C#, and Java applications. The SecureData Simple API allows users to include data protection functionality in their applications, and to communicate with the Key Management Server to obtain keys.

2.3 Product Description

The TOE provides its security functionality through implementation and management of the following features:

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

These features are described in more detail in the following subsections.

2.3.1 Identities

The Voltage 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 Voltage 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.

2.3.2 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.

2.3.3 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, which are excluded from the evaluated configuration).

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.

2.3.4 Formats

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.

2.3.5 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.

2.3.6 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.

2.4 Deployment Architecture

The TOE supports a number of different deployment architectures, but they can essentially be characterized as follows:

- A single-server system that includes all SDA components on one computer

- A multi-server system that distributes SDA components among at least two computers.

In a single-server system, the Management Console, Key Server, and Web Services Server are installed together on a single server. An example of a single server system that uses just the SecureData Simple API in a business application to connect to a database or some other type of storage would look similar to the following figure.

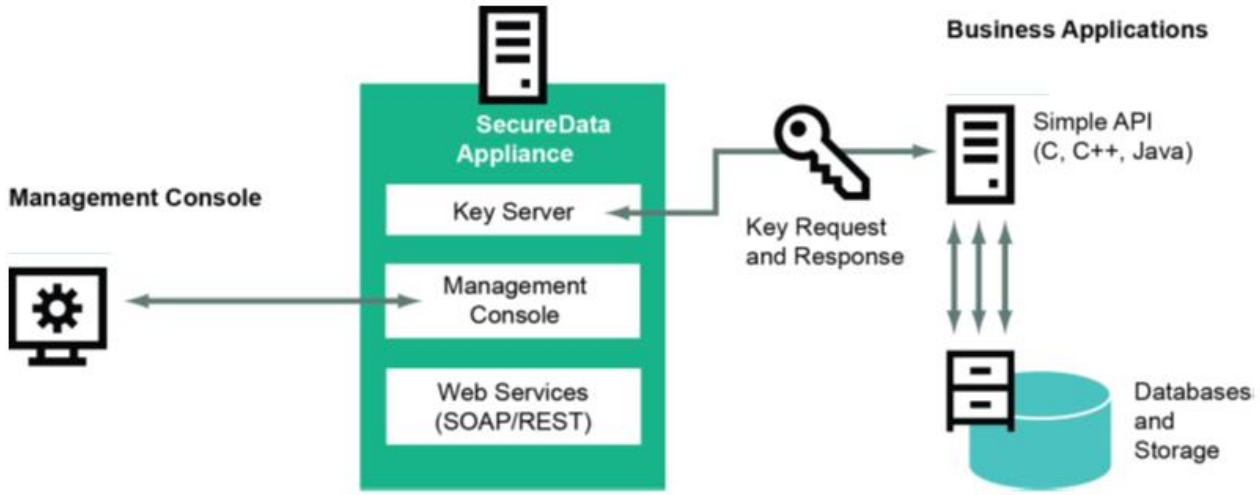

Figure 1: Example Single-Server Deployment

A multi-server system includes multiple servers that are managed from a single Management Console. In a multi-server system, the Management Console, Key Server, and Web Service Server can all be placed on separate servers. One server runs an active Management Console while one or more separate servers (termed *remote hosts*) run the Key Server and the Web Services Server. The configuration of the Voltage SecureData components can be optimized to suit customer requirements, including connections with load balancers to distribute client requests across remote hosts and configurations that provide redundant remote hosts, as depicted in the following figure. Further information and examples regarding multi-server deployments is available in *SecureData Architecture Guide*.

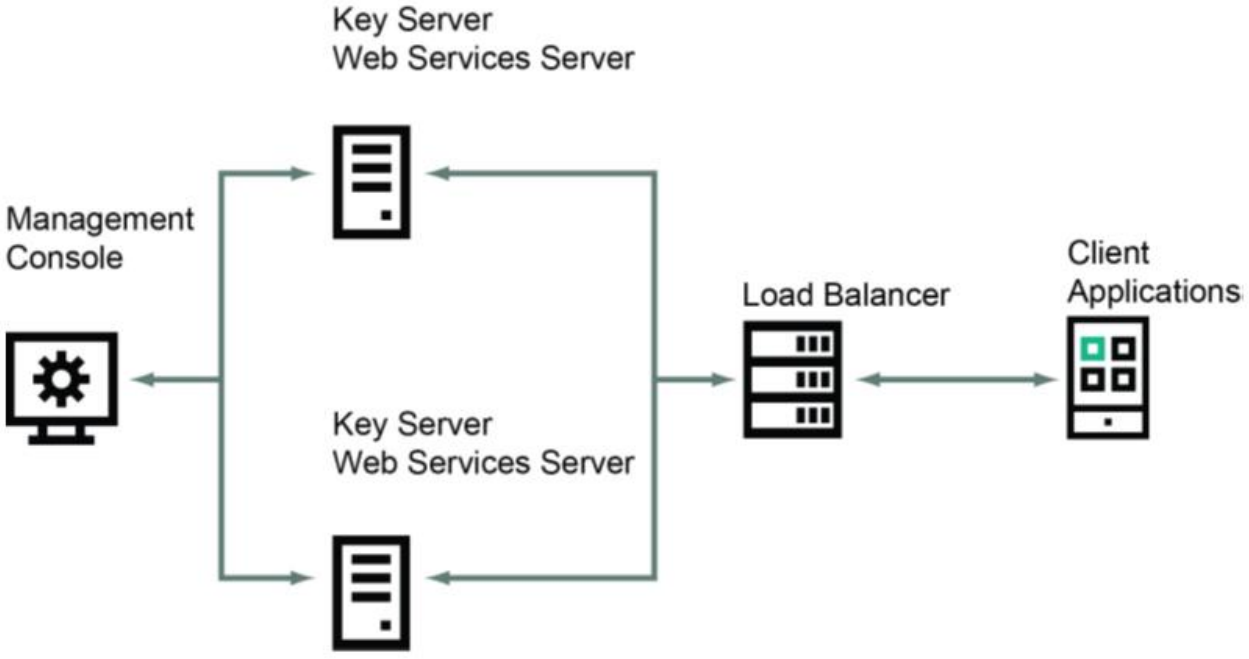

Figure 2: Example Multi-Server Deployment

2.5 Physical Boundaries

2.5.1 Physical TOE Components

The SDA is provisioned as the following ISO image that includes the SDA software and a hardened Linux-based operating system: `secure-data-appliance-6.4-227083.iso`.

The installation process installs the following components:

- Key Management Server
- Management Console
- Web Services Server (SOAP and REST APIs)
- KMS for Hadoop TDE (outside TOE boundary)
- Web Front End Server (outside TOE boundary).

The Voltage SecureData Simple API client software is provided in language-specific packages for the different platforms on which it is supported and is installed separately on client machines that require it to enable applications to communicate with and obtain key management services from the SDA.

2.5.2 Operational Environment Components

The TOE installation documentation specifies the following minimum hardware requirements for any server on which the SDA is to be installed:

- 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 minimum size; actual space required depends on logging levels and volume of data logged)
- Ability to load an ISO image containing the operating system, such as a physical or virtual DVD-ROM drive:
- 1 Gb Ethernet Network Interface Card.

The SDA is supported on all hardware compatible with Red Hat Enterprise Linux 7, 64 bit and on VMware ESXi Server 6.0.

The Simple API is supported on the various platforms specified in the following table.

Platform	Runtime	Languages
Windows XP SP3, 32 bit (deprecated) Windows Server 2003, 32 and 64 bit Windows Server 2008 SP1, 32 and 64 bit Windows Server 2012, 64 bit Windows 7, 32 and 64 bit Windows 8, 32 and 64 bit	JVM: 5 (deprecated), 6, 7, 8 .NET CLR: 4.0, 4.5, 4.6	C, Java, .NET
AIX 5.3 PowerPC, 32 and 64 bit	JVM: 5 (deprecated), 6	C, Java
AIX 6.1 PowerPC, 64 bit AIX 7.1 PowerPC, 64 bit	JVM: 5 (deprecated), 6, 7	C, Java
Solaris 10 SPARC, 32 and 64 bit	JVM: 5 (deprecated), 6, 7, 8	C, Java
HP-UX 11.23 Itanium, 32 and 64 bit	JVM: 5 (deprecated), 6, 7	C, Java
HP-UX 11.31 Itanium, 32 and 64 bit	JVM: 5 (deprecated), 6, 7, 8	C, Java

Platform	Runtime	Languages
Red Hat Enterprise Linux 5, 32 and 64 bit Red Hat Enterprise Linux 6, 64 bit Red Hat Enterprise Linux 7, 64 bit Ubuntu 12, 32 and 64 bit SUSE 11 - SP2, 64 bit	JVM: 5 (deprecated), 6, 7, 8	C, Java
Mac OS X 10.10 Yosemite, 64 bit	JVM: 6, 7, 8	C, Java

Table 1: SecureData Simple Client API Platform Support

The TOE supports the following optional components in its operational environment:

- NTP server to provide time synchronization to TOE platform
- LDAP server to support user authentication
- HSM to support storage of root keys.

2.6 Logical Boundaries

This section summarizes the security functions provided by the TOE.

2.6.1 Audit

The TOE is able to generate audit records of security relevant and other events as they occur, including: starting and stopping the audit function; all use of user identification and authentication mechanisms; and all use of management functions.

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.

Generated audit records are stored in a database and protected from unauthorized deletion. The audit records are digitally signed and the TOE is able to detect if stored audit records have been modified. The TOE provides authorized users with the capability to read audit information from the audit records. The audit records are displayed in a manner suitable for the authorized user to interpret the information. The TOE provides capabilities to search audit data for review based on specified search criteria and to select audit data for review based on time interval, audit record fields and specific values of selected fields.

2.6.2 Cryptographic Support

The TOE provides implementations of the following cryptographic capabilities: Format Preserving Encryption (FPE); embedded Format Preserving Encryption (eFPE); Identity-Based Encryption (IBE); and Identity-Based Symmetric Encryption (IBSE). In support of these capabilities, the TOE generates master keys from which encryption keys are derived and destroys master keys when they are no longer required.

2.6.3 User Data Protection

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.

2.6.4 Identification & Authentication

The users of the TOE comprise ***administrators***, who manage the TOE and its configuration, and ***clients***, who request key management services from the TOE.

The TOE supports the local (i.e., on device) definition of administrators with usernames and passwords and enforces minimum requirements for the construction of administrator passwords. Additionally, the TOE can be configured to use an LDAP directory to support remote user authentication. 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.

When the TOE receives client requests for encryption or decryption keys, 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; LDAP username and password; certificate.

2.6.5 Security Management

The SDA implements two management interfaces—the Appliance Menu, which is used for initial configuration of the SDA, and the Management Console, which provides the capabilities necessary to manage the TOE security functionality.

The TOE defines a single role (administrator) that is assigned to every user with an account on the TOE.

2.6.6 Protection of the TSF

When the TOE is configured as a multi-server system, communications between distributed components of the TOE occur over TLS, which provides confidentiality and integrity of transmitted data. In addition, the Simple API client software communicates with the SDA over TLS.

The SDA includes a Linux-based operating system that provides a reliable time stamp derived from the hardware clock of the computer on which the SDA software is installed. The system time can be set manually by an administrator via the Appliance Menu, or the SDA can be configured to synchronize its clock using NTP.

2.6.7 TOE Access

The TOE will terminate interactive sessions after 15 minutes of inactivity. The TOE also allows user-initiated termination of the user's own interactive session by explicitly logging off.

The TOE can be configured to limit remote access to IP addresses matching administrator-configured IP addresses or address patterns.

2.6.8 Trusted Path/Channels

The TOE provides a trusted channel to communicate securely with SecureData clients in the operational environment. The trusted channel is implemented using TLS, which ensures all communication over the channel is protected from disclosure and modification.

The TOE provides a trusted path for administrators to communicate with the TOE. The trusted path is implemented using HTTPS (i.e., TLS over HTTP) for access to the Management Console and SSH for remote access to the Appliance Menu. The trusted path is used for initial authentication and all subsequent administrative actions. The use of HTTPS and SSH ensures all communication over the trusted path is protected from disclosure and modification.

2.7 TOE Documentation

This section identifies the guidance documentation included in the TOE. The documentation comprises:

- SecureData Appliance Installation Guide, Version 6.4, October 2017
- SecureData Administrator Guide, Version 6.4, October 2017
- SecureData Architecture Guide, Version 6.4, October 2017
- SecureData REST API Developer Guide, Version 6.4, October 2017
- SecureData SOAP API Developer Guide, Version 6.4, October 2017
- SecureData Simple API Installation Guide, Version 5.10, May 2017
- SecureData Simple API Developer Guide, Version 5.10, May 2017.

3. Security Problem Definition

This section defines the security problem to be addressed by the TOE, in terms of threats to be countered by the TOE or its operational environment, and assumptions about the intended operational environment of the TOE.

3.1 Assumptions

This section contains assumptions regarding the operational environment and the intended usage of the TOE.

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 physical modification.

3.2 Threats

This section identifies and describes the threats to be countered by the TOE and its operational environment.

T.BRUTE_FORCE	An unauthorized user may gain access to the TOE through repeated password-guessing attempts.
T.DATA_COMPROMISE	Data on long-term storage media may be compromised if control of that media passes to unauthorized entities.
T.INTEGRITY_COMPROMISE	An unauthorized user may attempt to modify or destroy audit data, thus removing evidence of unauthorized or malicious activity.
T.KEY_COMPROMISE	Encrypted data may be compromised if unauthorized users gain access to encryption keys.
T.NETWORK_COMPROMISE	An unauthorized user may monitor the enterprise network in an attempt to obtain sensitive data, such as passwords, or to modify transmitted data.
T.NO_ACCOUNTABILITY	Authorized users of the TOE perform adverse actions on the TOE, or attempt to perform unauthorized actions, which go undetected.
T.UNATTENDED_SESSION	An unauthorized user gains access to the TOE via an unattended authorized user session.
T.UNAUTHORIZED_ACCESS	An unauthorized user may gain access to the TOE security functions and data.

4. Security Objectives

This section identifies the security objectives for the TOE and its operational environment. The security objectives identify the responsibilities of the TOE and its environment in addressing the security problem defined in Section 3.

4.1 Security Objectives for the TOE

The following are the TOE security objectives:

O.ACCESS_CONTROL	The TOE shall control access of Web Service clients to keys and key operations based on user name, authentication credentials, and IP address.
O.AUDIT	The TOE shall be able to generate audit records of security-relevant events.
O.AUDIT_REVIEW	The TOE shall provide a means for authorized users to review the audit records generated by the TOE.
O.CRYPTOGRAPHY	The TOE shall provide services for authorized users to request cryptographic operations using keys stored and managed by the TOE.
O.I_AND_A	The TOE shall require all users of the TOE to be identified and authenticated before gaining access to TOE services.
O.PASSWORD_CONTROLS	The TOE shall provide a mechanism to reduce the likelihood that users choose weak passwords.
O.PROTECTED_COMMS	The TOE shall protect communications between its distributed components and between itself and external entities.
O.SECURITY_MANAGEMENT	The TOE shall restrict the ability to perform security management functions on the TOE to authorized administrators having appropriate privileges.
O.SESSION_CONTROL	The TOE shall provide capabilities to deny session establishment based on IP address.
O.SESSION_TERMINATION	The TOE shall provide mechanisms to terminate a user session after a period of inactivity or at the request of the user.
O.STORAGE	The TOE shall protect stored audit records from unauthorized deletion and undetected modification.

4.2 Security Objectives for the Operational Environment

The following are the security objectives for the operational environment of the TOE.

OE.PHYSICAL	Those responsible for the TOE must ensure that those parts of the TOE critical to security policy are protected from any physical attack.
OE.PERSONNEL	Those responsible for the TOE must ensure that personnel working as authorized administrators have been carefully selected and trained for proper operation of the TOE.

5. IT Security Requirements

5.1 TOE Security Functional Requirements

This section specifies the security functional requirements (SFRs) for the TOE. SFRs were drawn from Part 2 of the Common Criteria v3.1 Revision 4.

Requirement Class	Requirement Component
FAU: 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
FCS: Cryptographic Support	FCS_CKM.1 – Cryptographic key generation
	FCS_CKM.4 – Cryptographic key destruction
	FCS_COP.1 – Cryptographic operation
FDP: User Data Protection	FDP_ACC.1 – Subset access control
	FDP_ACF.1 – Security attribute based access control
FIA: 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
FMT: Security Management	FMT_MSA.1 – Management of security attributes
	FMT_MSA.3 – Static attribute initialisation
	FMT_SMF.1 – Specification of Management Functions
	FMT_SMR.1 – Security roles
FPT: Protection of the TSF	FPT_ITT.1 – Basic internal TSF data transfer protection
	FPT_STM.1 – Reliable time stamps
FTA: TOE Access	FTA_SSL.3 – TSF-initiated termination
	FTA_SSL.4 – User-initiated termination
	FTA_TSE.1 – TOE session establishment
FTP: Trusted Path/Channels	FTP_ITC.1 – Inter-TSF trusted channel
	FTP_TRP.1 – Trusted path

Table 2: TOE Security Functional Components

5.1.1 Security Audit (FAU)

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 mechanism**
- **All use of the user authentication mechanism**
- **Use of the management functions**
- **Termination of an interactive session by the user**

].

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**].

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.

FAU_SAR.1 – Audit review

FAU_SAR.1.1 The TSF shall provide [**administrator**] 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.

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**

].

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.

5.1.2 Cryptographic Support (FCS)

FCS_CKM.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**].

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 bits**] that meet the following: [**RFC 5091: Identity-Based Cryptography Standard (IBCS) #1: Supersingular Curve Implementations of the BF and BB1 Cryptosystems**].

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**].

FCS_COP.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**] 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)**].

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-Boyen Identity Based Encryption**] and cryptographic key sizes [**3072 bits**] that meet the following: [**1363.3-2013 - IEEE Standard for Identity-Based Cryptographic Techniques using Pairings**].

5.1.3 User Data Protection (FDP)

FDP_ACC.1 – Subset access control

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

- **Subjects: Web Service Clients**
- **Objects: Keys**
- **Operations: protect; access; masked access**

].

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: [

- **Web Service Clients: user name; authentication credential; IP address**
- **Keys: identity**

].

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [

A Web Service Client can perform a requested operation on a Key if all of the following conditions are satisfied:

- **The Web Service Client's authentication credential matches the authentication criteria specified in an Identity Authorization rule**
- **The Web Service Client's user name matches a user name pattern in the same Identity Authorization rule**
- **The operation requested by the Web Service Client is enabled by the same Identity Authorization rule**
- **The Web Service Client's IP address matches an IP address pattern in an IP Authorization rule and the operation requested by the Web Service Client is enabled by the same IP Authorization rule**

].

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 deny rules**].

5.1.4 Identification and Authentication (FIA)

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**
- **Account Status**].

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 alphabetic character**].

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.

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.

].

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.

5.1.5 Security Management (FMT)

FMT_MSA.1 – Management of security attributes

FMT_MSA.1.1 The TSF shall enforce the [**SecureData Access Control SFP**] to restrict the ability to [***modify***] the security attributes [**Identity Authorization rules, IP Authorization rules**] to [**administrator**].

FMT_MSA.3 – Static attribute initialisation

FMT_MSA.3.1 The TSF shall enforce the [**SecureData Access Control SFP**] to provide [***restrictive***] 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.

FMT_SMF.1 – Specification of Management Functions

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**

].

FMT_SMR.1 – Security roles

FMT_SMR.1.1 The TSF shall maintain the roles: [**administrator**].

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

5.1.6 Protection of the TSF (FPT)

FPT_ITT.1 – Basic internal TSF data transfer protection

FPT_ITT.1.1 The TSF shall protect TSF data from [***disclosure, modification***] when it is transmitted between separate parts of the TOE.

FPT_STM.1 – Reliable time stamps

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps.

5.1.7 TOE Access (FTA)

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**].

FTA_SSL.4 – TSF-initiated termination

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

FTA_TSE.1 – TOE session establishment

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

5.1.8 Trusted Path/Channels (FTP)

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**].

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]***].

5.2 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the EAL 2 components as specified in Part 3 of the Common Criteria. No operations are applied to the assurance components.

Requirement Class	Requirement Component
ADV: Development	ADV_ARC.1 – Security architecture description
	ADV_FSP.2 – Security-enforcing functional specification
	ADV_TDS.1 – Basic design
AGD: Guidance documents	AGD_OPE.1 – Operational user guidance
	AGD_PRE.1 – Preparative procedures
ALC: Life-cycle support	ALC_CMC.2 – Use of a CM system
	ALC_CMS.2 – Parts of the TOE CM coverage
	ALC_DEL.1 – Delivery procedures
	ALC_FLR.1 – Basic flaw remediation
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
ATE: Tests	ATE_COV.1 – Evidence of coverage
	ATE_FUN.1 – Functional testing
	ATE_IND.2 – Independent testing – sample
AVA: Vulnerability assessment	AVA_VAN.2 – Vulnerability analysis

Table 3: TOE Security Assurance Components

5.2.1 Development (ADV)

ADV_ARC.1 – Security architecture description

ADV_ARC.1.1D The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.

ADV_ARC.1.2D The developer shall design and implement the TSF so that it is able to protect itself from tampering by untrusted active entities.

ADV_ARC.1.3D The developer shall provide a security architecture description of the TSF.

ADV_ARC.1.1C The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.

ADV_ARC.1.2C The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.

ADV_ARC.1.3C The security architecture description shall describe how the TSF initialization process is secure.

ADV_ARC.1.4C The security architecture description shall demonstrate that the TSF protects itself from tampering.

ADV_ARC.1.5C	The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.
ADV_ARC.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.2 – Security-enforcing functional specification

ADV_FSP.2.1D	The developer shall provide a functional specification.
ADV_FSP.2.2D	The developer shall provide a tracing from the functional specification to the SFRs.
ADV_FSP.2.1C	The functional specification shall completely represent the TSF.
ADV_FSP.2.2C	The functional specification shall describe the purpose and method of use for all TSFI.
ADV_FSP.2.3C	The functional specification shall identify and describe all parameters associated with each TSFI.
ADV_FSP.2.4C	For each SFR-enforcing TSFI, the functional specification shall describe the SFR-enforcing actions associated with the TSFI.
ADV_FSP.2.5C	For each SFR-enforcing TSFI, the functional specification shall describe direct error messages resulting from processing associated with the SFR-enforcing actions.
ADV_FSP.2.6C	The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.
ADV_FSP.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_FSP.2.2E	The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

ADV_TDS.1 – Basic design

ADV_TDS.1.1D	The developer shall provide the design of the TOE.
ADV_TDS.1.2D	The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.
ADV_TDS.1.1C	The design shall describe the structure of the TOE in terms of subsystems.
ADV_TDS.1.2C	The design shall identify all subsystems of the TSF.
ADV_TDS.1.3C	The design shall describe the behavior of each SFR-supporting or SFR non-interfering TSF subsystem in sufficient detail to determine that it is not SFR-enforcing.
ADV_TDS.1.4C	The design shall summarise the SFR-enforcing behavior of the SFR-enforcing subsystems.
ADV_TDS.1.5C	The design shall provide a description of the interactions among SFR-enforcing subsystems of the TSF, and between the SFR-enforcing subsystems of the TSF and other subsystems of the TSF.
ADV_TDS.1.6C	The mapping shall demonstrate that all TSFIs trace to the behavior described in the TOE design that they invoke.
ADV_TDS.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_TDS.1.2E	The evaluator shall determine that the design is an accurate and complete instantiation of all security functional requirements.

5.2.2 Guidance Documents (AGD)

AGD_OPE.1 – Operational user guidance

AGD_OPE.1.1D	The developer shall provide operational user guidance.
AGD_OPE.1.1C	The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.
AGD_OPE.1.2C	The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3C	The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.
AGD_OPE.1.4C	The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.
AGD_OPE.1.5C	The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.
AGD_OPE.1.6C	The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfil the security objectives for the operational environment as described in the ST.
AGD_OPE.1.7C	The operational user guidance shall be clear and reasonable.
AGD_OPE.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_PRE.1 – Preparative procedures

AGD_PRE.1.1D	The developer shall provide the TOE including its preparative procedures.
AGD_PRE.1.1C	The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.
AGD_PRE.1.2C	The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.
AGD_PRE.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AGD_PRE.1.2E	The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

5.2.3 Life-cycle Support (ALC)

ALC_CMC.2 – Use of a CM system

ALC_CMC.2.1D	The developer shall provide the TOE and a reference for the TOE.
ALC_CMC.2.2D	The developer shall provide the CM documentation.
ALC_CMC.2.3D	The developer shall use a CM system.
ALC_CMC.2.1C	The TOE shall be labelled with its unique reference.
ALC_CMC.2.2C	The CM documentation shall describe the method used to uniquely identify the configuration items.
ALC_CMC.2.3C	The CM system shall uniquely identify all configuration items.
ALC_CMC.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_CMS.2 – Parts of the TOE CM coverage

ALC_CMS.2.1D	The developer shall provide a configuration list for the TOE.
ALC_CMS.2.1C	The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; and the parts that comprise the TOE.
ALC_CMS.2.2C	The configuration list shall uniquely identify the configuration items.
ALC_CMS.2.3C	For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.
ALC_CMS.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_DEL.1 – Delivery procedures	
ALC_DEL.1.1D	The developer shall document and provide procedures for delivery of the TOE or parts of it to the consumer.
ALC_DEL.1.2D	The developer shall use the delivery procedures.
ALC_DEL.1.1C	The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.
ALC_DEL.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_FLR.1 – Basic flaw remediation	
ALC_FLR.1.1D	The developer shall document and provide flaw remediation procedures addressed to TOE developers.
ALC_FLR.1.1C	The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.
ALC_FLR.1.2C	The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.
ALC_FLR.1.3C	The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.
ALC_FLR.1.4C	The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.
ALC_FLR.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.4 Security Target Evaluation (ASE)

ASE_CCL.1 – Conformance claims	
ASE_CCL.1.1D	The developer shall provide a conformance claim.
ASE_CCL.1.2D	The developer shall provide a conformance claim rationale.
ASE_CCL.1.1C	The conformance claim shall contain a CC conformance claim that identifies the version of the CC to which the ST and the TOE claim conformance.
ASE_CCL.1.2C	The CC conformance claim shall describe the conformance of the ST to CC Part 2 as either CC Part 2 conformant or CC Part 2 extended.
ASE_CCL.1.3C	The CC conformance claim shall describe the conformance of the ST to CC Part 3 as either CC Part 3 conformant or CC Part 3 extended.
ASE_CCL.1.4C	The CC conformance claim shall be consistent with the extended components definition.
ASE_CCL.1.5C	The conformance claim shall identify all PPs and security requirement packages to which the ST claims conformance.
ASE_CCL.1.6C	The conformance claim shall describe any conformance of the ST to a package as either package-conformant or package-augmented.
ASE_CCL.1.7C	The conformance claim rationale shall demonstrate that the TOE type is consistent with the TOE type in the PPs for which conformance is being claimed.
ASE_CCL.1.8C	The conformance claim rationale shall demonstrate that the statement of the security problem definition is consistent with the statement of the security problem definition in the PPs for which conformance is being claimed.
ASE_CCL.1.9C	The conformance claim rationale shall demonstrate that the statement of security objectives is consistent with the statement of security objectives in the PPs for which conformance is being claimed.
ASE_CCL.1.10C	The conformance claim rationale shall demonstrate that the statement of security requirements is consistent with the statement of security requirements in the PPs for which conformance is being claimed.

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

ASE_ECD.1 – Extended components definition

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

ASE_INT.1 – ST introduction

ASE_INT.1.1D	The developer shall provide an ST introduction.
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 identify the TOE.
ASE_INT.1.4C	The TOE overview shall summarise the usage and major security features of the TOE.
ASE_INT.1.5C	The TOE overview shall identify the TOE type.
ASE_INT.1.6C	The TOE overview shall identify any non-TOE hardware/software/firmware required by the TOE.
ASE_INT.1.7C	The TOE description shall describe the physical scope of the TOE.
ASE_INT.1.8C	The TOE description shall describe the logical scope of the TOE.
ASE_INT.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ASE_INT.1.2E	The evaluator shall confirm that the TOE reference, the TOE overview, and the TOE description are consistent with each other.

ASE_OBJ.2 – Security objectives

ASE_OBJ.2.1D	The developer shall provide a statement of security objectives.
ASE_OBJ.2.2D	The developer shall provide a security objectives rationale.
ASE_OBJ.2.1C	The statement of security objectives shall describe the security objectives for the TOE and the security objectives for the operational environment.
ASE_OBJ.2.2C	The security objectives rationale shall trace each security objective for the TOE back to threats countered by that security objective and OSPs enforced by that security objective.
ASE_OBJ.2.3C	The security objectives rationale shall trace each security objective for the operational environment back to threats countered by that security objective, OSPs enforced by that security objective, and assumptions upheld by that security objective.
ASE_OBJ.2.4C	The security objectives rationale shall demonstrate that the security objectives counter all threats.

ASE_OBJ.2.5C	The security objectives rationale shall demonstrate that the security objectives enforce all OSPs.
ASE_OBJ.2.6C	The security objectives rationale shall demonstrate that the security objectives for the operational environment uphold all assumptions.
ASE_OBJ.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_REQ.2 – Derived security requirements

ASE_REQ.2.1D	The developer shall provide a statement of security requirements.
ASE_REQ.2.2D	The developer shall provide a security requirements rationale.
ASE_REQ.2.1C	The statement of security requirements shall describe the SFRs and the SARs.
ASE_REQ.2.2C	All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.
ASE_REQ.2.3C	The statement of security requirements shall identify all operations on the security requirements.
ASE_REQ.2.4C	All operations shall be performed correctly.
ASE_REQ.2.5C	Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.
ASE_REQ.2.6C	The security requirements rationale shall trace each SFR back to the security objectives for the TOE.
ASE_REQ.2.7C	The security requirements rationale shall demonstrate that the SFRs meet all security objectives for the TOE.
ASE_REQ.2.8C	The security requirements rationale shall explain why the SARs were chosen.
ASE_REQ.2.9C	The statement of security requirements shall be internally consistent.
ASE_REQ.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_SPD.1 – Security problem definition

ASE_SPD.1.1D	The developer shall provide a security problem definition.
ASE_SPD.1.1C	The security problem definition shall describe the threats.
ASE_SPD.1.2C	All threats shall be described in terms of a threat agent, an asset, and an adverse action.
ASE_SPD.1.3C	The security problem definition shall describe the OSPs.
ASE_SPD.1.4C	The security problem definition shall describe the assumptions about the operational environment of the TOE.
ASE_SPD.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_TSS.1 – TOE summary specification

ASE_TSS.1.1D	The developer shall provide a TOE summary specification.
ASE_TSS.1.1C	The TOE summary specification shall describe how the TOE meets each SFR.
ASE_TSS.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ASE_TSS.1.2E	The evaluator shall confirm that the TOE summary specification is consistent with the TOE overview and the TOE description.

5.2.5 Tests (ATE)

ATE_COV.1 – Evidence of coverage

ATE_COV.1.1D	The developer shall provide evidence of the test coverage.

ATE_COV.1.1C	The evidence of the test coverage shall show the correspondence between the tests in the test documentation and the TSFIs in the functional specification.
ATE_COV.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_FUN.1 – Functional testing

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

ATE_IND.2 – Independent testing – sample

ATE_IND.2.1D	The developer shall provide the TOE for testing.
ATE_IND.2.1C	The TOE shall be suitable for testing.
ATE_IND.2.2C	The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.
ATE_IND.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ATE_IND.2.2E	The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.
ATE_IND.2.3E	The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

5.2.6 Vulnerability Assessment (AVA)

AVA_VAN.2 – Vulnerability analysis

AVA_VAN.2.1D	The developer shall provide the TOE for testing.
AVA_VAN.2.1C	The TOE shall be suitable for testing.
AVA_VAN.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_VAN.2.2E	The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.
AVA_VAN.2.3E	The evaluator shall perform an independent vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design and security architecture description to identify potential vulnerabilities in the TOE.
AVA_VAN.2.4E	The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

6. TOE Summary Specification

This section describes the following security functions implemented by the TOE to satisfy the SFRs claimed in Section 5.1:

- Security Audit
- Cryptographic Support
- User Data Protection
- Identification and Authentication
- Security Management
- Protection of the TSF
- TOE Access
- Trusted Path/Channels.

6.1 Security Audit

The 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 generated audit records are digitally signed and are stored in a database where they are protected from unauthorized deletion. The digital signature applied to each audit record enables the TOE to determine if an audit record stored in the database has been modified.

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 security function satisfies the following security functional requirements:

- FAU_GEN.1—the TOE generates audit records for security relevant events that include the date and time of the event, type of event, subject identity, and outcome of the event.
- FAU_GEN.2—the TOE associates each auditable event resulting from actions of identified users with the identity of the user that caused the event.
- FAU_SAR.1—the TOE provides administrators with the capability to read audit information from the audit records. The audit records are displayed in a manner suitable for the administrator to interpret the information.
- FAU_SAR.3—the TOE provides capabilities to search audit data for review based on specified search criteria and to select displayed records based on time interval, specification of audit record fields, and specific values of selected fields.
- FAU_STG.1—the TOE protects stored audit records from unauthorized deletion and is able to detect any modification of stored audit records.

6.2 Cryptographic Support

The TOE provides implementations of the following cryptographic capabilities:

- Format Preserving Encryption (FPE)—encrypts data so that the ciphertext has the same length and character set as the input data. This capability uses AES in FF1 mode with 128, 192 or 256 bit keys, as defined in NIST SP 800-38G *Recommendation for Block Cipher Modes of Operation: Methods for Format-Preserving Encryption*.
- 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. This capability also uses AES in FF1 mode with 128, 192 or 256 bit keys, as defined in NIST SP 800-38G.
- Identity-Based Encryption (IBE)—an asymmetric algorithm that encrypts data without preserving its length or character set. This capability supports Boneh-Franklin Identity Based Encryption (IBE BF) and Boneh-Boyen Identity Based Encryption (IBE BB1) identity-based encryption schemes using 3072 bit keys (as defined in IEEE 1363.3-2013 – *Standard for Identity-Based Cryptographic Techniques using Pairings*).
- Identity-Based Symmetric Encryption (IBSE)—encrypts data without preserving its length or character set. This capability uses AES in CBC mode (as defined in NIST SP 800-38A *Recommendation for Block Cipher Modes of Operation: Methods and Techniques*) or AES in EME* mode (as defined in IEEE P1619.2 – *Standard for Wide-Block Encryption for Shared Storage Media*). IBSE supports 128, 192 or 256 bit keys.

In support of these capabilities, the TOE generates master keys from which encryption keys are derived and destroys master keys when they are no longer required. The TOE generates master keys for each of the supported encryption capabilities when it creates a district. This includes AES keys for each of the capabilities that use an AES mode of operation and private keys for the IBE BF and IBE BB1 modes of identity based encryption.

The TOE destroys master keys by overwriting with zeroes anytime a district is deleted.

The Cryptographic Support security function satisfies the following security functional requirements:

- FCS_CKM.1(1)—the TOE generates 128, 192 and 256 bit keys used with AES, using a hash-based DRBG that meets NIST SP 800-90A.
- FCS_CKM.1(2)—the TOE generates 3072 bit private keys used with IBE BF and IBE BB1 in accordance with RFC 5091.
- FCS_CKM.4—the TOE destroys cryptographic keys by overwriting with zeroes.
- FCS_COP.1(1)—the TOE performs symmetric encryption and decryption with AES in CBC mode, AES in FF1 mode, and IBSE using AES in CBC or EME* mode, using key sizes of 128, 192 and 256 bits.
- FCS_COP.1(2)—the TOE performs asymmetric encryption and decryption with Boneh-Franklin Identity Based Encryption and Boneh-Boyen Identity Based Encryption, using key sizes of 3072 bits.

6.3 User Data Protection

The TOE is able to restrict the operations available to clients using the Voltage 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.

6.3.1 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. Note that this does not deny access to the client—only that this rule does not apply for accessing data. If there is another rule that grants the client masked or full access, that rule is considered
    - 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.

6.3.2 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. Note that this does not deny access to the client—only that this rule does not apply for accessing data. If there is another rule that grants the client masked or full access, that rule is considered

- 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, access, or masked access operations on data, based on the client's identity, authentication credentials, and IP address.

6.4 Identification and Authentication

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

6.4.1 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.

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.

6.4.2 Client I&A

When the TOE receives requests for encryption or decryption keys from Voltage 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
- Remote LDAP Authentication—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. Client certificate authentication must be specifically enabled for the Simple API on the SDA and the appropriate trusted root certificate must be present on the SDA so that the passed client certificate can be verified.

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; authentication data; and account status (enabled or disabled).
- 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 supports multiple authentication mechanisms: local password; remote LDAP authentication; shared secret; and client certificate.
- FIA_UID.2—the TOE requires each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

6.5 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 Micro Focus – Voltage 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 Voltage 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 **Web Service** tab of the Management Console GUI provides administrators with the capability to manage the Identity Authorization and IP Authorization rules used in enforcing the SecureData Access Control SFP on Web Service clients. When either an Identity Authorization or IP Authorization rule is created, the default setting disables Protect operations and specifies “No Access” for the Access Level. At the time of rule creation, the administrator can enable the Protect operation and specify an alternative Access Level.

The Security Management function satisfies the following security functional requirements:

- FMT_MSA.1—the TOE provides the capabilities for administrators to manage the Identity Authorization and IP Authorization rules used in enforcing the SecureData Access Control SFP on Web Service clients.
- FMT_MSA.3—the TOE provides restrictive default values for Identity Authorization and IP Authorization rules. The administrator can specify alternative initial values to override the defaults.
- 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.

6.6 Protection of the TSF

When the TOE is configured as a multi-server system, communications between distributed components of the TOE occur over TLS, which provides confidentiality and integrity of transmitted data. In addition, the Simple API client software communicates with the SDA over TLS.

The SDA includes a Linux-based operating system that provides a reliable time stamp derived from the hardware clock of the computer on which the SDA software is installed. The system time can be set manually by an administrator via the Appliance Menu, or the SDA can be configured to synchronize its clock using NTP.

The Protection of the TSF security function satisfies the following security functional requirements:

- FPT_ITT.1—the TOE uses TLS to protect TSF data from disclosure and modification when it is transmitted between distributed parts of the TOE.
- FPT_STM.1—the TOE is able to provide reliable time stamps, based on a time source in its operational environment.

6.7 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.

6.8 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., HTTP over TLS) 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, 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, either directly using the SDA server's console, or remotely using SSH. SSH provides a trusted path connection for the administrator remotely accessing the Appliance Menu.

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 and SSH to access the Appliance Menu.

7. Rationale

This section provides the rationale for completeness and consistency of the Security Target. The rationale addresses the following areas:

- Security Objectives
- Security Functional Requirements
- Security Assurance Requirements
- Requirement Dependencies
- TOE Summary Specification.

7.1 Security Objectives Rationale

This section shows that all secure usage assumptions and threats are completely covered by security objectives for the TOE or operational environment. In addition, each objective counters or addresses at least one assumption or threat.

	T.BRUTE_FORCE	T.DATA_COMPROMISE	T.INTEGRITY_COMPROMISE	T.KEY_COMPROMISE	T.NETWORK_COMPROMISE	T.NO_ACCOUNTABILITY	T.UNATTENDED_SESSION	T. UNAUTHORIZED_ACCESS	A. MANAGE	A. PROTECT
O.ACCESS_CONTROL				X						
O.AUDIT						X				
O.AUDIT_REVIEW						X				
O.CRYPTOGRAPHY		X								
O.I_AND_A								X		
O.PASSWORD_CONTROLS	X									
O.PROTECTED_COMMS					X					
O.SECURITY_MANAGEMENT				X				X		
O.SESSION_CONTROL								X		
O.SESSION_TERMINATION							X			
O.STORAGE			X							
OE.PHYSICAL										X
OE.PERSONNEL									X	

Table 4: Security Problem Definition to Security Objective Correspondence

T.BRUTE_FORCE

An unauthorized user may gain access to the TOE through repeated password-guessing attempts.

This threat is countered by the following security objective:

- O.PASSWORD_CONTROLS—addresses this threat by providing a mechanism, configurable by an administrator, which encourages users to choose difficult-to-guess passwords.

T.DATA_COMPROMISE

Data on long-term storage media may be compromised if control of that media passes to unauthorized entities.

This threat is countered by the following security objectives:

- O.CRYPTOGRAPHY—addresses this threat by providing services that clients can request for encrypting and decrypting data to be stored long-term.

T.INTEGRITY_COMPROMISE

An unauthorized person may attempt to modify or destroy audit data, thus removing evidence of unauthorized or malicious activity.

This threat is countered by the following security objective:

- O.STORAGE—addresses this threat by ensuring the TOE is able to protect stored audit records from unauthorized deletion and undetected modification.

T.KEY_COMPROMISE

Encrypted data may be compromised if unauthorized users gain access to encryption keys.

This threat is countered by the following security objectives:

- O.ACCESS_CONTROL— addresses this threat by ensuring access to keys managed by the TOE is restricted to authorized and authenticated users.
- O.SECURITY_MANAGEMENT—supports O.ACCESS_CONTROL in addressing this threat by providing administrators the capabilities to configure and manage access control.

T.NETWORK_COMPROMISE

An unauthorized user may monitor the enterprise network in an attempt to obtain sensitive data, such as passwords, or to modify transmitted data.

This threat is countered by the following security objective:

- O.PROTECTED_COMMS—addresses this threat by ensuring the TOE is able to protect communications between its distributed components and between itself and external entities.

T.NO_ACCOUNTABILITY

Authorized users of the TOE perform adverse actions on the TOE, or attempt to perform unauthorized actions, which go undetected.

This threat is countered by the following security objectives:

- O.AUDIT—addresses this threat by ensuring the TOE is able to generate audit records of security relevant events.
- O.AUDIT_REVIEW—supports O.AUDIT in addressing the threat by ensuring the TOE provides capabilities for effective review of stored audit records.

T.UNATTENDED_SESSION

An unauthorized user gains access to the TOE via an unattended authorized user session.

This threat is countered by the following security objectives:

- O.SESSION_TERMINATION—addresses this threat by providing users with a mechanism to terminate their interactive sessions with the TOE, and by ensuring sessions that have been inactive for a configurable period of time will be terminated by the TOE.

T.UNAUTHORIZED_ACCESS

An unauthorized user may gain access to the TOE security functions and data.

This threat is countered by the following security objective:

- O.I_AND_A—addresses this threat by ensuring all users of the TOE are identified and authenticated prior to gaining further access to the TOE and its services.
- O.SECURITY_MANAGEMENT—supports O.I_AND_A in addressing this threat by providing administrators the capabilities to configure and manage user authentication.
- O.SESSION_CONTROL—supports O.I_AND_A in addressing this threat by ensuring attempts to establish a session with the TOE can be controlled and restricted to approved IP addresses.

A.MANAGE

There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.

This assumption is satisfied by the following security objective:

- OE.PERSONNEL—this objective satisfies the assumption by ensuring those assigned as authorized administrators are properly trained in operating the TOE.

A.PROTECT

The TOE hardware and software critical to the security policy enforcement will be located within controlled access facilities which will prevent unauthorized physical access.

This assumption is satisfied by the following security objective:

- OE.PHYSICAL—this objective satisfies the assumption by ensuring the TOE is protected from physical attack.

7.2 Security Functional Requirements Rationale

All security functional requirements identified in this Security Target are fully addressed in this section and each is mapped to the objective it is intended to satisfy. Table 5 summarizes the correspondence of functional requirements to TOE security objectives.

	O.ACCESS_CONTROL	O.AUDIT	O.AUDIT_REVIEW	O.CRYPTOGRAPHY	O.I_AND_A	O.PASSWORD_CONTROLS	O.PROTECTED_COMMS	O.SECURITY_MANAGEMENT	O.SESSION_CONTROL	O.SESSION_TERMINATION	O.STORAGE
FAU_GEN.1		X									
FAU_GEN.2		X									
FAU_SAR.1			X								
FAU_SAR.3			X								
FAU_STG.1											X
FCS_CKM.1(*)				X							
FCS_CKM.4				X							
FCS_COP.1(*)				X							

	O.ACCESS_CONTROL	O.AUDIT	O.AUDIT_REVIEW	O.CRYPTOGRAPHY	O.I_AND_A	O.PASSWORD_CONTROLS	O.PROTECTED_COMMS	O.SECURITY_MANAGEMENT	O.SESSION_CONTROL	O.SESSION_TERMINATION	O.STORAGE
FDP_ACC.1	X										
FDP_ACF.1	X										
FIA_ATD.1					X						
FIA_SOS.1						X					
FIA_UAU.2					X						
FIA_UAU.5					X						
FIA_UID.2					X						
FMT_MSA.1	X										
FMT_MSA.3	X										
FMT_SMF.1								X			
FMT_SMR.1								X			
FPT_ITT.1							X				
FPT_STM.1		X									
FTA_SSL.3										X	
FTA_SSL.4										X	
FTA_TSE.1									X		
FTP_ITC.1							X				
FTP_TRP.1							X				

Table 5: Objectives to Requirement Correspondence

O.ACCESS_CONTROL

The TOE shall control access of Web Service clients to keys and key operations based on user name, authentication credentials, and IP address.

The following security functional requirements contribute to satisfying this security objective:

- FDP_ACC.1, FDP_ACF.1—the ST includes FDP_ACC.1 and FDP_ACF.1 to specify the access control policy enforced by the TOE to control access to key objects to authorized Web Service clients.
- FMT_MSA.1, FMT_MSA.3—the ST includes FMT_MSA.1 and FMT_MSA.3 to specify restrictions on the management of security attributes used to control access to key objects.

O.AUDIT

The TOE shall be able to generate audit records of security-relevant events.

The following security functional requirements contribute to satisfying this security objective:

- FAU_GEN.1—the ST includes FAU_GEN.1 to specify the capability to generate audit records of security-relevant events, and to specify the specific events to be audited and the content of generated audit records of those events.
- FAU_GEN.2—the ST supports FAU_GEN.1 by including FAU_GEN.2 to specify the capability to record the identity of users associated with audited events.

- FPT_STM.1—the ST supports FAU_GEN.1 by including FPT_STM.1 to specify the capability to provide reliable time stamps, which are applied to generated audit records.

O.AUDIT_REVIEW

The TOE shall provide a means for authorized users to review the audit records generated by the TOE.

The following security functional requirements contribute to satisfying this security objective:

- FAU_SAR.1—the ST includes FAU_SAR.1 to specify which roles are to be able to read data from stored audit records.
- FAU_SAR.3—the ST supports FAU_SAR.1 by including FAU_SAR.3 to specify capabilities for searching audit records based on specified search criteria and selecting audit records based on time interval, audit record field, and specific values of selected fields.

O.CRYPTOGRAPHY

The TOE shall provide services for authorized users to request cryptographic operations using keys stored and managed by the TOE.

The following security functional requirements contribute to satisfying this security objective:

- FCS_COP.1(*)—the ST includes FCS_COP.1(*) to specify the cryptographic operations that authorized users can request from the TOE.
- FCS_CKM.1(*)—the ST supports FCS_COP.1(*) by including FCS_CKM.1(*) to specify the keys that the TOE can generate in support of the cryptographic operations specified by FCS_COP.1(*).
- FCS_CKM.4—the ST includes FCS_CKM.4 to specify the capability to destroy cryptographic keys associated with a district when the district is deleted.

O.I_AND_A

The TOE shall require all users of the TOE to be identified and authenticated before gaining access to TOE services.

The following security functional requirements contribute to satisfying this security objective:

- FIA_UID.2, FIA_UAU.2—the ST includes FIA_UID.2 and FIA_UAU.2 to specify that users must be successfully identified and authenticated by the TOE before being able to perform any other TSF-mediated actions.
- FIA_ATD.1—the ST supports FIA_UID.2 and FIA_UAU.2 by including FIA_ATD.1 to ensure user identity and authentication data security attributes are associated with individual users.
- FIA_UAU.5—the ST supports FIA_UAU.2 by including FIA_UAU.5 to specify the authentication mechanisms supported by the TOE and the rules by which the TOE authenticates a user's claimed identity.

O.PASSWORD_CONTROLS

The TOE shall provide a mechanism to reduce the likelihood that users choose weak passwords.

The following security functional requirement contributes to satisfying this security objective:

- FIA_SOS.1—the ST includes FIA_SOS.1 to specify that passwords must meet minimum construction requirements, in terms of length and character set.

O.PROTECTED_COMMS

The TOE shall protect communications between its distributed components and between itself and external entities.

The following security functional requirements contribute to satisfying this security objective:

- FPT_ITT.1—the ST includes FPT_ITT.1 to specify that TSF data communicated between distributed parts of the TOE will be protected from disclosure and modification.
- FTP_ITC.1—the ST includes FTP_ITC.1 to specify that data will be communicated between the TOE and external IT entities through a trusted channel that protects the data from disclosure and modification.
- FTP_TRP.1—the ST includes FTP_TRP.1 to specify that data will be communicated between the TOE and remote users through a trusted path that protects the data from disclosure and modification.

O.SECURITY_MANAGEMENT

The TOE shall provide the security management functions necessary to support the cryptographic services provided by the TOE.

The following security functional requirements contribute to satisfying this security objective:

- FMT_SMF.1, FMT_SMR.1—the ST includes these requirements to specify the security management functions to be provided by the TOE (FMT_SMF.1) and the supported security management roles (FMT_SMR.1).

O.SESSION_CONTROL

The TOE shall provide capabilities to deny session establishment based on IP address.

The following security functional requirement contributes to satisfying this security objective:

- FTA_TSE.1—the ST includes FTA_TSE.1 to specify the capability for the TSF to deny establishment of an interactive session with the TOE based on the requester's IP address.

O.SESSION_TERMINATION

The TOE shall provide mechanisms to terminate a user session after a period of inactivity or at the request of the user.

The following security functional requirements contribute to satisfying this security objective:

- FTA_SSL.3—the ST includes FTA_SSL.3 to specify the capability for the TSF to terminate an interactive user session after a period of inactivity.
- FTA_SSL.4—the ST includes FTA_SSL.4 to specify the capability for users to terminate their own interactive sessions.

O.STORAGE

The TOE shall protect stored audit records from unauthorized deletion and undetected modification.

The following security functional requirements contribute to satisfying this security objective:

- FAU_STG.1—the ST includes FAU_STG.1 to specify that stored audit records will be protected from unauthorized deletion and that the TSF will be able to detect unauthorized modification of stored audit records.

7.3 Security Assurance Requirements Rationale

EAL 2 was selected as the assurance level because the TOE is a commercial product whose users require a low to moderate level of independently assured security. The TOE is intended for use in an environment with good physical access security where it is assumed that attackers will have Basic attack potential. The target assurance level of EAL 2 is appropriate for such an environment.

7.4 Requirement Dependency Rationale

The following table identifies the SFRs claimed in the ST, their dependencies as defined in CC Part 2, and how the dependency is satisfied in the ST. It can be seen that all dependencies have been satisfied by inclusion in the ST of the appropriate dependent SFRs.

Requirement	Dependencies	How Satisfied
FAU_GEN.1	FPT_STM.1	FPT_STM.1
FAU_GEN.2	FAU_GEN.1, FIA_UID.1	FAU_GEN.1, FIA_UID.2
FAU_SAR.1	FAU_GEN.1	FAU_GEN.1
FAU_SAR.3	FAU_SAR.1	FAU_SAR.1
FAU_STG.1	FAU_GEN.1	FAU_GEN.1
FCS_CKM.1(*)	[FCS_CKM.2 or FCS_COP.1], FCS_CKM.4	FCS_COP.1(*), FCS_CKM.4
FCS_CKM.4	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	FCS_CKM.1
FCS_COP.1(*)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4	FCS_CKM.1(*), FCS_CKM.4
FDP_ACC.1	FDP_ACF.1	FDP_ACF.1
FDP_ACF.1	FDP_ACC.1, FMT_MSA.3	FDP_ACC.1, FMT_MSA.3
FIA_ATD.1	None	None
FIA_SOS.1	None	None
FIA_UAU.2	FIA_UID.1	FIA_UID.2
FIA_UAU.5	None	None
FIA_UID.2	None	None
FMT_MSA.1	[FDP_ACC.1 or FDP_IFC.1], FMT_SMR.1, FMT_SMF.1	FDP_ACC.1, FMT_SMR.1, FMT_SMF.1
FMT_MSA.3	FMT_MSA.1, FMT_SMR.1	FMT_MSA.1, FMT_SMR.1
FMT_SMF.1	None	None
FMT_SMR.1	FIA_UID.1	FIA_UID.2
FPT_ITT.1	None	None
FPT_STM.1	None	None
FTA_SSL.3	None	None
FTA_SSL.4	None	None
FTA_TSE.1	None	None
FTP_ITC.1	None	None
FTP_TRP.1	None	None

Table 6: Requirement Dependencies

7.5 TOE Summary Specification Rationale

Section 6, the TOE Summary Specification, describes how the security functions of the TOE meet the claimed SFRs. The following table provides a mapping of the SFRs to the security function descriptions to support the TOE Summary Specification.

	Security Audit	Cryptographic Support	User Data Protection	Identification and Authentication	Security Management	Protection of the TSF	TOE Access	Trusted Path/Channels
FAU_GEN.1	X							
FAU_GEN.2	X							
FAU_SAR.1	X							
FAU_SAR.3	X							
FAU_STG.1	X							
FCS_CKM.1(*)		X						
FCS_CKM.4		X						
FCS_COP.1(*)		X						
FDP_ACC.1			X					
FDP_ACF.1			X					
FIA_ATD.1				X				
FIA_SOS.1				X				
FIA_UAU.2				X				
FIA_UAU.5				X				
FIA_UID.2				X				
FMT_MSA.1					X			
FMT_MSA.3					X			
FMT_SMF.1					X			
FMT_SMR.1					X			
FPT_ITT.1						X		
FPT_STM.1						X		
FTA_SSL.3							X	
FTA_SSL.4							X	
FTA_TSE.1							X	
FTP_ITC.1								X
FTP_TRP.1								X

Table 7: Security Functions vs. Requirements Mapping