Sigaba SigabaNet 2.2 Security Target

Version 4.0

January 7, 2006

Final

Prepared for:

Secure Data in Motion, Incorporated, dba Sigaba®
1875 S. Grant Street
San Mateo, CA 94402
www.sigaba.com

Prepared by:

Ashton Security Laboratories
12530 Rock Ridge Road
Herndon, VA 20170
www.ashtonlabs.com

1 Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is provided by Secure Data in Motion, Inc. (dba Sigaba). The product is the SigabaNet system; the TOE is a subset of the SigabaNet system, specifically the SigabaNet Authentication Server and the SigabaNet Key Server. The TOE components provide secure storage of cryptographic keys and controlled access to those keys. The TOE is intended for use with application programming interfaces outside of the TOE boundary that are given access to cryptographic keys stored by the TOE based on the Sigaba authentication mechanism, name assertions.

The Security Target contains the following additional sections:

- **TOE Description – Section 2**
- **Security Environment – Section 3**
- **Security Objectives – Section 4**
- **IT Security Requirements – Section 5**
- **TOE Summary Specification – Section 6**
- **Protection Profile Claims – Section 7**
- **Rationale – Section 8)**

1.1 Security Target, TOE, and CC Identification

- **ST Title** – Sigaba SigabaNet 2.2 Security Target
- **ST Version** – 4.0
- **ST Date** – January 7, 2006
- **TOE Identification** – Sigaba SigabaNet 2.2
- **CC Identification** – Common Criteria for Information Technology Security Evaluation, Version 2.1, August 1999, ISO/IEC 15408.

1.2 Conformance Claims

This ST is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation **Part 2**: Security functional requirements, **Version 2.1**, August 1999, ISO/IEC 15408-2.
  - **Part 2 Extended with FAU_EXP.1, an explicitly stated requirement for the audit function.**
- Common Criteria for Information Technology Security Evaluation **Part 3**: Security assurance requirements, **Version 2.1**, August 1999, ISO/IEC 15408-3.
  - **Part 3**
  - **EAL2, Augmented with ADV_SPM.1 to meet security functional requirement dependency**

1.3 Conventions, Terminology, Acronyms

This section specifies the formatting information used in the Security Target.

1.3.1 Conventions

The following conventions have been applied in this document:

- **Security Functional Requirements** – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement. Application notes are added to provide additional information about the implementation of a particular requirement.
    - **Iteration** allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a letter placed at the end of the component. For example FDP_ACC.1a and FDP_ACC.1b indicate that the ST includes two iterations of the FDP_ACC.1 requirement, a and b.
    - **Assignment** allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]).
    - **Selection**: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded 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 …").
    - **Application Notes** provide additional information about the implementation of a requirement and are in *italics*.
- **Other sections of the ST** – Other sections of the ST use bolding to highlight text of special interest, such as captions

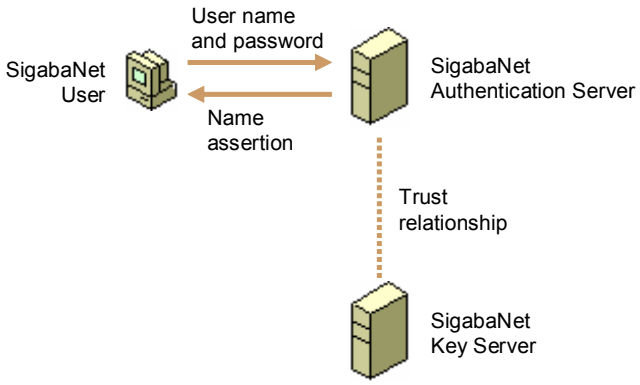
2 TOE Description

The TOE is the Key Server and Authentication Server components of the SigabaNet 2.2 product. The TOE is a sensitive data protection type of product that mediates access to cryptographic keys that are used to guard against unauthorized access to data. The TOE is intended for use with application programming interfaces outside of the TOE boundary that use the TOE to generate server credentials called name assertions on behalf of SigabaNet users. SigabaNet users (users of the SigabaNet product that includes the TOE) can then use these credentials for authentication within the SigabaNet system (a superset of the TOE), allowing them to generate, store, and manage secret keys for use by external client applications.

2.1 TOE Overview

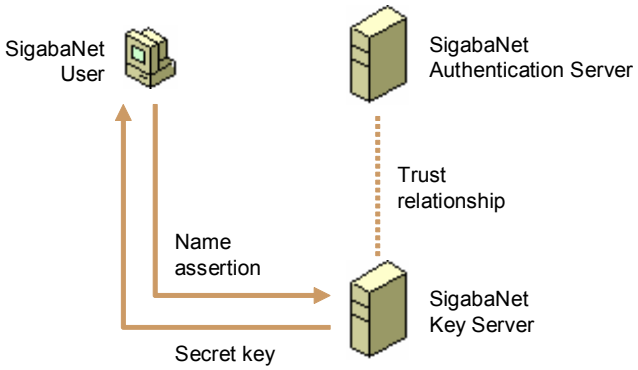
The SigabaNet Authentication server component generates credentials called name assertions for SigabaNet users or user applications outside the TOE. These name assertions can then be used for authenticating to the Key Server to gain access to cryptographic keys. A name assertion is an XML document that is compliant with the S2ML standard. It contains the name and email address of the authenticated individual along with entitlements (access rights) to data managed by the Key Server.

Figure 1: Creating name assertions



The SigabaNet Key Server component generates, stores, and manages secret keys for SigabaNet users for use by client applications outside of the TOE boundary. Secret keys are cryptographic keys that should not be made public that are used with symmetric key cryptographic algorithms. Secret keys provided by the SigabaNet TOE are uniquely associated with resources in the SigabaNet system that may be accessed by one or more SigabaNet users.

Figure 2: Creating secret keys using name assertions



2.2 TOE Architecture

The components that make up the TOE are:

- **SigabaNet Authentication and Key Servers** – SigabaNet Authentication and Key Server applications, each implemented as Java servlets
- **SigabaNet Authentication and Key Server files** – Compiled SigabaNet Authentication and Key Server servlet bytecode are stored in operating system files. A number of operating system files and Sigaba files are also used by the SigabaNet Authentication and Key Servers for configuration. Note that the web based administrative forms are used to update configuration files. The trusted administrators are relied upon to manage the TOE code and the TOE configuration files securely.

The underlying Windows 2000 Server SP4 operating system, Postgres 7.1.3 database server, Apache Tomcat version 4.1.24 application server, the SigabaNet Administration Server, and the SigabaNet Client APIs are all part of the IT environment (i.e. are all outside of the TOE boundary).

The following terms will be used in referring to the TOE, its components and its environment:

- SigabaNet product: The SigabaNet product as sold to customers. The TOE consists of just the Authentication and Key Server components of the SigabaNet product.
- SigabaNet system: The SigabaNet product in operation.
- SigabaNet Authentication Server: Part of the TOE.
- SigabaNet Key Server: Part of the TOE.
- SigabaNet Client: Part of the TOE Environment
- SigabaNet Administration Server: Part of the TOE environment

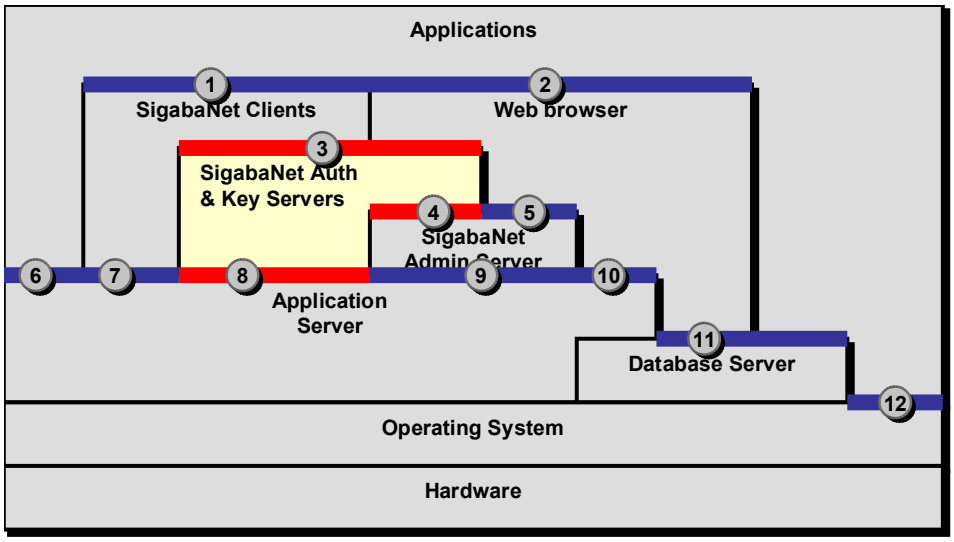
2.2.1 Physical Boundaries

There are several mechanisms to communicate with the SigabaNet TOE's Authentication and Key Servers:

- **S2ML and ESRP protocols**
- **HTTP and HTTPS protocols**
- **JMS publish/subscribe messages**

The S2ML and ESRP protocols are used to request SigabaNet Authentication and Key Server services using TCP/IP network connections. They are used by untrusted client processes, via routines in provided libraries, to communicate with the two servers. Administrators manage the SigabaNet Authentication and Key Servers using web browsers to access HTML pages with server configuration forms generated by SigabaNet Authentication and Key Server Java Server Pages (JSP) interfaces. Administrators also manage the SigabaNet Authentication and Key Servers by sending configuration data using Java Message System (JMS) messages.

Figure 3: SigabaNet system interfaces (numbered), including TOE interfaces



The interfaces to each Sigaba system component:

1. **Java API:** This interface consists of unspecified Windows applications running in the IT environment making SigabaNet client API library calls (not part of the TOE) to request SigabaNet Authentication and Key Server subcomponent services using authentication protocol and key sever network protocol interfaces, respectively.
2. **Win32 GUI:** This interface is the Windows graphical user interface of a web browser that is running in the IT environment.
3. **Network protocol:** This interface consists of the authentication protocol and key sever network protocol interfaces to SigabaNet Authentication and Key Server subcomponents, respectively.
4. **Java JMS:** This interface consists of Java Messaging Service (JMS) publish/subscribe network protocol interfaces.
5. **Network protocol:** This interface is the HTTP/HTTPS network protocol interface between a web browser and the administrative web interface of the SigabaNet Administration Server subcomponent.
6. **Win32:** This interface consists of operating system calls made by unspecified Windows applications running in the IT environment.
7. **Java Servlet or Virtual Machine:** This interface between Java language calls made by SigabaNet client API calls in either a web browser JVM or the application server JVM.
8. **Java Servlet:** This interface consists of J2EE servlet calls made by SigabaNet Authentication and Key Server subcomponents. Note that all SigabaNet subcomponents access the network using application server JVM JNI TCP/IP sockets implementations.
9. **Java Servlet:** This interface consists of J2EE servlet calls made by SigabaNet Administration Server subcomponents. Note that all SigabaNet subcomponents access the network using application server JVM JNI TCP/IP sockets implementations.
10. **Network protocol:** This interface is the HTTP/HTTPS network protocol interface between a web browser and the administrative web interface of the application server.
11. **Network protocol:** This interface is the HTTP/HTTPS network protocol interface between a web browser and the administrative web interface of the database server.
12. **Win32:** Same as interface labeled #6 above. This interface consists of operating system calls made by unspecified Windows applications running in the IT environment.

The Sigaba Authentication subcomponent and the Sigaba Key Server subcomponent each implement interfaces labeled **3, 4, and 8** (i.e. network protocol interfaces such as ESRP, Java JMS network protocol interfaces, and Java servlet JVM call interfaces) as identified above. The remaining interfaces are interfaces

between components of the IT environment, which include: web browser, application server, database server, hardware.

2.2.2 Logical Boundaries

The TOE logically supports the following security functions at its interfaces:

- **Security audit**
- **Cryptographic support**
- **User data protection**
- **Identification and authentication**
- **Security management**

2.2.2.1 Security audit

The TOE has an audit mechanism that is invoked for access checks, authentication attempts, administrator functions, and at other times during its operation. When invoked, the date, time, responsible individual and other details describing the event are recorded to the audit trail.

The audit log is stored in a database in the environment. Audit events are generated by the SigabaNet Authentication and Key Servers using JMS publish messages, which are sent to SigabaNet system components that are in the environment and not part of the TOE.

2.2.2.2 Cryptographic support

The TOE implements a cryptomodule that was designed to meet FIPS 140-1 requirements. The TOE uses asymmetric keys to use with name assertions, and generates symmetric keys to use to protect data. The TOE provides pseudo random number generation, signature generation, signature verification, and hash generation. The TOE also provides cryptographic key generation.

2.2.2.3 User data protection

The TOE implements a Discretionary Access Control (DAC) policy over applicable SigabaNet system objects, specifically the secret keys. Each secret key object has an owner (the creator of the object). Object owners have special permissions, even though other users can subsequently be granted access to the object.

2.2.2.4 Identification and authentication

The TOE provides its own identification and authentication mechanism. Users must provide a valid user name and password before obtaining a name assertion, which in turn must be provided before either generating or accessing a secret key. Once identified and authenticated, all subsequent actions associated with that user and policy decisions are based on the user's identity, role, and name assertion entitlements.

2.2.2.5 Security management

The TOE provides administrators with HTML web forms that are generating using Java Server Pages (JSP) that in turn either operate directly on server configuration files or that generate Java Message Service (JMS) messages that are then sent to the server being administered. Administrator web forms are used to manage users and associated attributes, and JMS messages along with web forms are used to manage other security functions, including TSF data such as trust points. Web form interfaces are restricted to administrators only. Note that the TOE provides non-administrative users with network protocol interfaces that are accessed using application programming interfaces that are outside of the TOE boundary.

2.2.3 Functions of the IT Environment

The TOE relies on the application server to provide separate user connection threads in its JVM JNI TCP/IP sockets service. The SigabaNet TOE relies on the operating system to provide protection of the TSF by restricting access to TOE files. The operating system also provides a reliable time stamp. The SigabaNet TOE relies on SigabaNet system components that are in the environment and not part of the

TOE to receive audit events generated by SigabaNet Authentication and Key Servers, provide a reporting capability, and write events to an audit trail.

2.3 TOE Documentation

Sigaba provides documents that describe the installation process for the SigabaNet system (including the TOE) as well as guidance for subsequent use and administration of the applicable security features. Refer to Section 6 for information about these and other documentation associated with the SigabaNet system.

3 Security Environment

3.1 Threats

This security target has derived all security objectives from the statement of Organizational Security Policy found in the following section. Therefore, there is no statement of the explicit threats countered by this security target.

3.2 Organizational Security Policies

An Organizational Security Policy is a set of rules or procedures imposed by an organization upon its operations to protect its sensitive data. Although the organizational security policies described below is drawn from DoD Manual 5200.28-M (Techniques and procedures for Implementing, Deactivating and Evaluating Resource Sharing ADP Systems) it applies to many non-DoD environments.

P.AUTHORIZED_USERS	Only those users who have been authorized to access the information within the system may access the TOE.
P.NEED_TO_KNOW	The TOE must limit the access to, modification of, and destruction of the information in protected resources to those authorized users which have a “need to know” for that information.
P.ACCOUNTABILITY	The users of the TOE shall be held accountable for their actions within the system.

3.3 Secure Usage Assumptions

This section describes the security aspects of the environment in which the TOE will be, or is intended to be used. This includes information about the physical, personnel, and connectivity aspects of the environment.

The TOE is assured to provide effective security measures in a cooperative non-hostile environment only if it is installed, managed, and used correctly. The operational environment must be managed in accordance with assurance requirements documentation for delivery, operation, and user/administrator guidance. The following specific conditions are assumed to exist in an environment where the TOE is employed.

3.3.1 Physical Assumptions

The TOE is intended for application in user areas that have physical control and monitoring. It is assumed that the following physical conditions will exist:

A.LOCATE	The processing resources of the TOE will be located within controlled access facilities which will prevent unauthorized physical access.
A.PROTECT	The TOE hardware and software critical to security policy enforcement will be protected from unauthorized physical modification.

3.3.2 Personnel Assumptions

It is assumed that the following personnel conditions will exist:

A.MANAGE	There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.
A.NO_EVIL_ADM	The system administrative personnel are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the administrator documentation.
A.COOP	Authorized users possess the necessary authorization to access at least some of the information managed

by the TOE and are expected to act in a cooperating manner in a benign environment.

3.3.3 Connectivity Assumptions

This security target contains no explicit network or distributed system requirements. However, it is assumed that the following connectivity conditions exist:

A.PEER	Any other systems with which the TOE communicates are assumed to be under the same management control and operate under the same security policy constraints. Conformant TOEs are applicable to networked or distributed environments only if the entire network operates under the same constraints and resides within a single management domain. There are no security requirements which address the need to trust external systems or the communications links to such systems.
A.CONNECT	All connections to peripheral devices reside within the controlled access facilities. Conformant TOEs only address security concerns related to the manipulation of the TOE through its authorized access points. Internal communication paths to access points such as terminals are assumed to be adequately protected.

4 Security Objectives

4.1 Security Objectives for the TOE

O.ACCESS	The TOE will ensure that users gain only authorized access to it and to the resources that it controls.
O.ADMIN_ROLE_TOE	The TOE will provide authorized administrator roles to isolate administrative actions.
O.AUDIT_GENERATION	The TOE will provide the capability to detect and create records of security relevant events associated with users.
O.CRYPTO_ALGORITHMS	The TOE will implement approved cryptographic algorithms for signature generation and verification.
O.CRYPTO_KEYS	The TOE will generate cryptographic keys when requested by an authorized user.
O.DISCRETIONARY_ACCESS	The TOE will control access to resources based upon the identity of users.
O.MANAGE_TOE	The TOE will provide functions and facilities necessary to support the authorized administrators in their management of the security of the TOE.
O.USER_AUTHENTICATION	The TOE will verify the claimed identity of users.
O.USER_IDENTIFICATION	The TOE will uniquely identify users.

4.2 Security Objectives for the Environment

4.2.1 Security Objectives for the IT Environment

OE.AUDIT_ON_OFF	The IT Environment will generate an audit record when TOE auditing is turned off or on.
OE.AUDIT_PROTECTION	The IT Environment will provide the capability to protect audit information.
OE.AUDIT_REVIEW	The IT Environment will provide the capability to selectively view audit information.
OE.TIME	The IT environment will provide a time source that provides reliable time stamps.
OE.TOE_PROTECTION_IT	The IT Environment will protect the TOE and its assets from external interference or tampering.

4.2.2 Security Objectives for the Non-IT Environment

Certain objectives with respect to the general operating environment must be met. The following are the non-IT security objectives:

OE.INSTALL	Those responsible for the TOE must ensure that the TOE is delivered, installed, managed, and operated in a manner which maintains IT security objectives.
OE.PHYSICAL	Those responsible for the TOE must ensure that those parts of the TOE critical to security policy are protected from physical attack which might compromise IT security objectives.

OE.CREDEN	Those responsible for the TOE must ensure that all access credentials, such as passwords or other authentication information, are protected by the users in a manner which maintains IT security objectives.

5 IT Security Requirements

This section defines the security functional and security assurance requirements for the TOE and associated IT environment components. Note that in addition to these requirements, the SigabaNet TOE also satisfies a minimum strength of function 'SOF-basic. The only applicable (i.e., probabilistic or permutational) security functions are FIA_SOS.1, FIA_UAU.2, and FIA_UID.2, which are all levied on the TOE.

5.1 TOE Security Functional Requirements

The following table describes the SFRs that are candidates to be satisfied by the SigabaNet TOE. Note that all of the SFRs are CC Part II conformant, with the exception of those requirements marked as "(Explicitly stated requirement)" in the table below.

Requirement class	Requirement component
FAU: Security audit	FAU_EXP.1: Audit data generation explicitly stated (Explicitly stated requirement) FAU_GEN.2: User identity association
FCS: Cryptographic support	FCS_COP.1: Cryptographic operation FCS_CKM.1: Cryptographic key generation FCS_CKM.2: Cryptographic key distribution FCS_CKM.4: Cryptographic key destruction
FDP: User data protection	FDP_ACC.2: Complete access control FDP_ACF.1: Security attribute based access control
FIA: Identification and authentication	FIA_UAU.2: User authentication before any action FIA_UID.2: User identification before any action
FMT: Security management	FMT_MSA.1: Management of security attributes FMT_MSA.2: Secure security attributes FMT_MSA.3: Static attribute initialization FMT_SMF.1: Specification of management functions FMT_SMR.1: Security roles

5.1.1 Security audit (FAU)

5.1.1.1 Audit data generation explicitly stated (FAU_EXP.1)

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

a) Start-up of the audit functions;

b) The following auditable events:

i. Success in creating a new name assertion

ii. Success in creating a new secret key

iii. All requests to retrieve an existing secret key.

FAU_EXP 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, authentication realm.

Application Note: The audit function is always on for the TOE, i.e., there is no mechanism to turn auditing on or off; if the TOE is running, then auditing is on. The TOE does generate an audit record for turning on the TOE, and therefore for turning on auditing, but there is no audit record generated for turning off the TOE and auditing. Since the TOE auditing function does not include generation of an audit record for turning off auditing, the TOE does not meet the requirement of FAU_GEN.1 and FAU_EXP1 is explicitly stated instead to define the TOE auditing function. FAU_GEN.1 is specified in the IT Environment, since turning the TOE off and on, and therefore turning TOE auditing off and on, is recorded by the Operating System and these audit records are stored in a Windows audit log.

5.1.1.2 User identity association (FAU_GEN.2)

FAU_GEN.2.1 The TSF shall be able to associate each auditable event with the identity of the user that caused the event.

5.1.2 Cryptographic operation (FCS)

5.1.2.1 Cryptographic operation (FCS_COP.1)

FCS_COP.1.1 The TSF shall perform **[pseudo random number generation, signature generation, signature verification, and hash generation]** in accordance with a specified cryptographic algorithm **[listed below]** and cryptographic key sizes **[specified for each algorithm]** that meet the following: **[standards noted for each algorithm].**

a.) Pseudo random number generation in accordance with FIPS 186-2 Appendix 3
b.) Signature generation/verification RSA 1024, 2048 (PKCS #1, FIPS PUB 186-2, and ANSI X9.31) with SHA-1 (FIPS PUB 180-1, ANSI X9.30 Part 2), DSA 1024 bits (FIPS PUB 186-2, ANSI X9.30) with SHA-1
c.) Hash generation SHA-1 (FIPS PUB 180-1, ANSI X9.30 Part 2)

5.1.2.2 Cryptographic key generation (FCS_CKM.1)

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with the specified cryptographic key generation algorithm **[listed below]** and specified cryptographic key sizes **[specified for each algorithm]** that meet the following: **[standards noted for each algorithm]**

a.) RSA 1024, 2048 bits key pairs in accordance with PKCS #1
b.) 3DES 112, 168 bits (ANSI X9.52).
c.) DSA 1024 bits key pairs in accordance with FIPS PUB 186-2, ANSI X9.30

5.1.2.3 Cryptographic key distribution (FCS_CKM.2)

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method **[SigabaNet custom method]** that meets the following: **[none]**.

5.1.2.4 Cryptographic key destruction (FCS_CKM.4)

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

5.1.3 User data protection (FDP)

5.1.3.1 Complete access control (FDP_ACC.2)

FDP_ACC.2.1 The TSF shall enforce the **[Discretionary Access Control (DAC) SFP]** on **[**

d.) **Subjects: Threads acting on behalf of users,**
e.) **Objects: Secret keys]**

and all operations among subjects and objects covered by the SFP.

FDP_ACC.2.2 The TSF shall ensure that all operations between any subject in the TSC and any object within the TSC are covered by an access control SFP.

5.1.3.2 Security attribute based access control (FDP_ACF.1)

FDP_ACF.1.1 The TSF shall enforce the **[DAC SFP]** to objects based on the following: **[**

a.) **The user identity and authentication realm associated with a subject; and**

b.) **The following access control attributes associated with an object:**

iv. **Object creator, and**

v. **Object Access Control List (ACL) (ACLs can be used to grant or deny access to the granularity of a single user using ACL entries that include user name and authentication realm)]**.

(per International Interpretation #103)

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.) **If a user presents a valid name assertion and entitlement, and if the entitlement authorizes the use of the TOE subcomponent that is enforcing the DAC SFP, then:**

i. **If the subject is the creator of the object, access is granted.**

ii. **If the subject has been granted access according to an entry in the object ACL, access is granted]**.

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based in the following additional rules: **[none]**.

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the **[none]**.

5.1.4 Identification and authentication (FIA)

5.1.4.1 User authentication before any action (FIA_UAU.2)

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

5.1.4.2 User identification before any action (FIA_UID.2)

FIA_UID.2.1 The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on the behalf of that user.

5.1.5 Security management (FMT)

5.1.5.1 Management of security attributes (FMT_MSA.1)

FMT_MSA.1.1 The TSF shall enforce the **[DAC SFP]** to restrict the ability to **[*modify*]** the security attributes **[object ACL]** to **[**

a.) **Object creator,**

b.) **System administrator]**.

5.1.5.2 Secure security attributes (FMT_MSA.2)

FMT_MSA.2.1 The TSF shall ensure that only secure values are accepted for security attributes.

5.1.5.3 Static attribute initialisation (FMT_MSA.3)

FMT_MSA.3.1 The TSF shall enforce the **[DAC 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 **[Object creator]** to specify alternative initial values to override the default values when an object or information is created.

5.1.5.4 Specification of management functions (FMT_SMF.1)

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

a.) **Create objects**

b.) **Modify object ACL**

c.) **Specify alternative initial values for objects**

(per International Interpretation #65)

5.1.5.5 Security roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles **[System Administrator and User]**

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

5.2 IT Environment Security Functional Requirements

The following table describes the SFRs that are satisfied by the IT environment of the SigabaNet TOE.

Requirement class	Requirement component
FAU: Security audit	FAU_GEN.1: Audit data generation FAU_STG.1: Protected audit trail storage FAU_SAR.1: Audit review
FPT: Protection of the TSF	FPT_RVM.1: Non-bypassability of the TSP FPT_SEP.1: TSF domain separation FPT_STM.1: Reliable time stamps

5.2.1 Security audit (FAU)

5.2.1.1 Audit data generation (FAU_GEN.1)

FAU_GEN.1.1 The ~~TSF~~ **IT Environment** 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: no other events]**.

(per International Interpretation #202)

FAU_GEN 1.2 The ~~TSF~~ **IT Environment** shall record within each audit record at least the following information:

c.) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and

d.) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, **[no other audit relevant information].**

Application Note: The OS records turning on the TOE and turning off the TOE, which equates to turning on auditing within the TOE and turning off auditing within the TOE.

5.2.1.2 Protected audit trail storage (FAU_STG.1)

FAU_STG.1.1 The ~~TSF~~ **IT environment** shall protect the stored audit records from unauthorized deletion.

FAU_STG.1.2 The ~~TSF~~ **IT environment** shall be able to **[*prevent*]** unauthorized modifications to the audit records in the audit trail. *(per International Interpretations #141 and #202)*

5.2.1.3 Audit review (FAU_SAR.1)

FAU_SAR.1.1 The ~~TSF~~ **IT environment** shall provide **[System Administrator]** with the capability to read **[all audit information]** from the audit records.

FAU_SAR.1.2 The ~~TSF~~ **IT environment** shall provide the audit records in a manner suitable for the user to interpret the information.

5.2.2 Protection of the TSF (FPT)

5.2.2.1 Non-bypassability of the TSP (FPT_RVM.1)

FPT_RVM.1.1 The ~~TSF~~ **IT environment** shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

5.2.2.2 TSF domain separation (FPT_SEP.1)

FPT_SEP.1.1 The ~~TSF~~ **IT environment** shall maintain a security domain for ~~its own~~ TSF execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.1.2 The ~~TSF~~ **IT environment** shall enforce separation between the security domains of subjects in the TSC.

5.2.2.3 Reliable time stamps (FPT_STM.1)

FPT_STM.1.1 The ~~TSF~~ **IT environment** shall be able to provide reliable time stamps for its own use.

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the EAL2 as specified in Part 3 of the Common Criteria, augmented by ADV_SPM.1, which is included because it is a dependency of one of the functional requirements, FMT_MSA.2. No operations are applied to the assurance components.

Requirement Class	Requirement Component
ACM: Configuration management	ACM_CAP.2: Configuration items
ADO: Delivery and operation	ADO_DEL.1: Delivery procedures
ADO: Delivery and operation	ADO_IGS.1: Installation, generation, and start-up procedures
ADV: Development	ADV_FSP.1: Informal functional specification
ADV: Development	ADV_HLD.1: Descriptive high-level design
ADV: Development	ADV_RCR.1: Informal correspondence demonstration
AGD: Guidance documents	AGD_ADM.1: Administrator guidance
AGD: Guidance documents	AGD_USR.1: User guidance
ATE: Tests	ATE_COV.1: Evidence of coverage
ATE: Tests	ATE_FUN.1: Functional testing
ATE: Tests	ATE_IND.2: Independent testing - sample
AVA: Vulnerability assessment	AVA_SOF.1: Strength of TOE security function evaluation
AVA: Vulnerability assessment	AVA_VLA.1: Developer vulnerability analysis

5.3.1 Configuration management (ACM)

5.3.1.1 Configuration items (ACM_CAP.2)

ACM_CAP.2.1D The developer shall provide a reference for the TOE.
ACM_CAP.2.2D The developer shall use a CM system.
ACM_CAP.2.3D The developer shall provide CM documentation.
ACM_CAP.2.1C The reference for the TOE shall be unique to each version of the TOE.
ACM_CAP.2.2C The TOE shall be labelled with its reference.
ACM_CAP.2.3C The CM documentation shall include a configuration list.
ACM_CAP.2.4C The configuration list shall describe the configuration items that comprise the TOE.
ACM_CAP.2.4C The configuration list shall uniquely identify all configuration items that comprise the TOE. (*per International Interpretation #3)*
ACM_CAP.2.5C The CM documentation shall describe the method used to uniquely identify the configuration items.
ACM_CAP.2.6C The CM system shall uniquely identify all configuration items.
ACM_CAP.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.2 Delivery and operation (ADO)

5.3.2.1 Delivery procedures (ADO_DEL.1)

ADO_DEL.1.1D The developer shall document procedures for delivery of the TOE or parts of it to the user.
ADO_DEL.1.2D The developer shall use the delivery procedures.
ADO_DEL.1.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.
ADO_DEL.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.2.2 Installation, generation, and start-up procedures (ADO_IGS.1)

ADO_IGS.1.1D The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.
ADO_IGS.1.1C The installation, generation and start-up documentation shall describe the steps necessary for secure installation, generation, and start-up of the TOE. (*per International Interpretation #51)*
ADO_IGS.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADO_IGS.1.2E The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

5.3.3 Development (ADV)

5.3.3.1 Informal functional specification (ADV_FSP.1)

ADV_FSP.1.1D The developer shall provide a functional specification.
ADV_FSP.1.1C The functional specification shall describe the TSF and its external interfaces using an informal style.
ADV_FSP.1.2C The functional specification shall be internally consistent.
ADV_FSP.1.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.
ADV_FSP.1.4C The functional specification shall completely represent the TSF.

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

ADV_FSP.1.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.2 Descriptive high-level design (ADV_HLD.1)

ADV_HLD.1.1D The developer shall provide the high-level design of the TSF.

ADV_HLD.1.1C The presentation of the high-level design shall be informal.

ADV_HLD.1.2C The high-level design shall be internally consistent.

ADV_HLD.1.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV_HLD.1.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV_HLD.1.5C The high-level design shall identify any underlying hardware, firmware, and/ or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV_HLD.1.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV_HLD.1.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

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

ADV_HLD.1.2E The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.3 Informal TOE Security Policy Model (ADV_SPM.1) [Dependency of FMT_MSA.2]

ADV_SPM.1.1D The developer shall provide a TSP model.

ADV_SPM.1.2D The developer shall demonstrate correspondence between the functional specification and the TSP model.

5.3.3.4 Informal correspondence demonstration (ADV_RCR.1)

ADV_RCR.1.1D The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.

ADV_RCR.1.1C For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

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

5.3.4 Guidance documents (AGD)

5.3.4.1 Administrator guidance (AGD_ADM.1)

AGD_ADM.1.1D The developer shall provide administrator guidance addressed to system administrative personnel.

AGD_ADM.1.1C The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.

AGD_ADM.1.2C The administrator guidance shall describe how to administer the TOE in a secure manner.

AGD_ADM.1.3C The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD_ADM.1.4C The administrator guidance shall describe all assumptions regarding user behaviour that are relevant to secure operation of the TOE.

AGD_ADM.1.5C The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

AGD_ADM.1.6C The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.
AGD_ADM.1.7C The administrator guidance shall be consistent with all other documentation supplied for evaluation.
AGD_ADM.1.8C The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.
AGD_ADM.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4.2 User guidance (AGD_USR.1)

AGD_USR.1.1D The developer shall provide user guidance.
AGD_USR.1.1C The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.
AGD_USR.1.2C The user guidance shall describe the use of user-accessible security functions provided by the TOE.
AGD_USR.1.3C The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.
AGD_USR.1.4C The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behaviour found in the statement of TOE security environment.
AGD_USR.1.5C The user guidance shall be consistent with all other documentation supplied for evaluation.
AGD_USR.1.6C The user guidance shall describe all security requirements for the IT environment that are relevant to the user.
AGD_USR.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.5 Tests (ATE)

5.3.5.1 Evidence of coverage (ATE_COV.1)

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 identified in the test documentation and the TSF as described 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.

5.3.5.2 Functional testing (ATE_FUN.1)

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, test procedure descriptions, expected test results and actual test results.
ATE_FUN.1.2C The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.
ATE_FUN.1.3C The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.
ATE_FUN.1.4C The expected test results shall show the anticipated outputs from a successful execution of the tests.
ATE_FUN.1.5C The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.
ATE_FUN.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.5.3 Independent testing - sample (ATE_IND.2)

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 test a subset of the TSF as appropriate to confirm that the TOE operates as specified.
ATE_IND.2.3E The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

5.3.6 Vulnerability assessment (AVA)

5.3.6.1 Strength of TOE security function evaluation (AVA_SOF.1)

AVA_SOF.1.1D The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim.
AVA_SOF.1.1C For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.
AVA_SOF.1.2C For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST.
AVA_SOF.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_SOF.1.2E The evaluator shall confirm that the strength claims are correct.

5.3.6.2 Developer vulnerability analysis (AVA_VLA.1)

AVA_VLA.1.1D The developer shall perform a vulnerability analysis. (*per International Interpretation #51)*
AVA_VLA.1.2D The developer shall provide vulnerability analysis documentation. (*per International Interpretation #51)*
AVA_VLA.1.1C The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for obvious ways in which a user can violate the TSP . (*per International Interpretation #51)*
AVA_VLA.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_VLA.1.2E The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure obvious vulnerabilities have been addressed.

6 TOE Summary Specification

6.1 TOE Security Functions

6.1.1 Security audit

The TOE has an audit mechanism that is invoked for access checks, authentication attempts, and the other events in the bulleted list below. When invoked, the date, time, responsible individual and other details describing the event are recorded to the audit trail. All audit records generated by actions undertaken on behalf of a user, include the identity of that user.

Each audit record identifies the event type, the responsible user, the authentication realm, the date and time of the event, an indication of success or failure, and other information specific to each audit event.

Auditable events include:

- **Success in creating a new name assertion**
- **Success in creating a new secret key**
- **All requests to retrieve an existing secret key**

This security function is designed to satisfy the following security functional requirements:

- **FAU_EXP.1** Audit records are generated for the events described above and audit records include the content described above.
- **FAU_GEN.2** A user identity is associated with each audit event that involves a user.

6.1.2 Cryptographic support

The TOE implements a crypto-module that was designed to meet FIPS 140-1 requirements. The TOE uses asymmetric keys to use with name assertions, and generates symmetric keys to use to protect data. The FIPS 140-1 *Security Requirements for Cryptographic Modules* document is available from the National Institute of Standards and Technology website at http://csrc.nist.gov/cryptval/140-1.htm.

This security function is designed to satisfy the following security functional requirements:

- **FCS_COP.1 –** The TOE provides pseudo random number generation, signature generation, signature verification, and hash generation. The pseudo random number generator is seeded with values generated during the user identification and authentication protocol exchange.
- **FCS_CKM.1 –** The TOE generates message keys in response to requests from users to the Key Server.
- **FCS_CKM.2** – The TOE distributes the requested message keys to users.
- **FCS_CKM.4** – The TOE destroys name assertion signature public and private keys when directed by the administrator. The TOE destroys user message keys when they expire.

6.1.3 User data protection

The TOE implements a Discretionary Access Control (DAC) policy over applicable TOE objects – secret keys. Each secret key object has an owner, who is the creator of the object. Object owners have special permissions, while other uses can subsequently be granted access to the object.

The TOE implements a DAC policy for object access based on:

- **User identity**
- **Authentication realm**
- **ACL**

The TOE objects subject to this policy are secret keys. Note that name assertions are not objects in the TOE, they are security attributes that belong to individual users.

The TOE stores access permissions for the applicable object in access control lists (ACLs) and provides the ability to grant and revoke access permissions. The user identity and authentication realm membership are used by the DAC mechanism to validate the user's permission to access the applicable object.

When a user in the TOE creates an object, that user becomes the object owner. In general the owner of an object has all access permissions to the object regardless of explicitly granted or revoked access permissions.

This security function is designed to satisfy the following security functional requirements:

- **FDP_ACC.2** All TOE users are subject to the DAC policy for all available operations on secret keys.
- **FDP_ACF.1** Secret keys have owners and ACLs and these attributes are compared against user identities in order to determine whether the request operation should be allowed. If a check fails, access is denied.

6.1.4 Identification and authentication

The TOE provides its own identification and authentication mechanism. Users must provide a valid user name and password before obtaining a name assertion, which in turn must be provided before either generating or accessing a secret key. Once identified and authenticated, all subsequent actions associated with that user, including the application of policy, are based on the user's identity, role, and name assertion entitlements.

To login to the TOE, the user provides the user name, password, and authentication realm to the SigabaNet Authentication Server. The SigabaNet Authentication Server hashes the password and compares the resulting value to that stored in the configured database in the environment, and then checks membership in the authentication realm. If either the password or authentication realm check fails, the login request will fail and no functions will be made available.

As a result of a successful login, a subject is created on behalf of the client and is represented by a unique Java thread.

In addition to the user's identifiers and password, any role assigned to the user is also stored in the database in the environment.

This security function is designed to satisfy the following security functional requirements:

- **FIA_UAU.2** The TOE offers no TSF-mediated functions until the user is authenticated.
- **FIA_UID.2** The TOE offers no TSF-mediated functions until the user is identified.

6.1.5 Security management

The TOE provides administrators with HTML web forms that are generated using Java Server Pages (JSP) that in turn either operate directly on server configuration files or generate Java Message Service (JMS) messages that are then sent to the server being administered. Administrator web forms are used to manage users and associated attributes directly, and JMS messages sent from web forms are used to manage other security functions, including TSF data such as trust points. Web form interfaces are restricted to access by administrators only. Note that the TOE provides non-administrative users with network protocol interfaces that are accessed using application programming interfaces that are outside of the TOE boundary.

The TOE maintains a number of tables containing configuration data in a database in the environment, as well as a number of configuration files to control how it operates. For example, database tables contain user account information such as authentication data (i.e., hashed passwords) and role membership, while configuration files contain administrative web interface configuration data.

This security function is designed to satisfy the following security functional requirements:

- **FMT_SMF.1** The TOE provides the following security management functions: creation of objects, modification of object ACLs, and specification of alternative initial values for objects.
- **FMT_MSA.1** The ability to modify object attributes in the ACL is restricted by the DAC SFP to the object creator and the system administrators.

- **FMT_MSA.2** The TOE accepts only secure security attributes by generating the name assertion that has access to a key maintained by the key server.
- **FMT_MSA.3** By default every object is created with restrictive values in its ACL. Specifically, the creator is set as the owner and no other access is provided. Owner access grants the ability to read and modify the object and to manage object attributes (i.e., change access values in the ACL). Subsequently, the owner or a system administrator can grant access to other users by adding entries to an object's ACL.
- **FMT_SMR.1** Each user account can be assigned either a system administrator or user role.

6.2 TOE Security Assurance Measures

6.2.1 Configuration management

These activities are documented in:

- **SigabaNet 2.2 Configuration Management** – Provides a reference for the TOE that is unique to the version of the TOE. Also provides a configuration list that describes the configuration items that comprise the TOE. Note that items in the configuration list are uniquely identified including a description of the underlying CM system.

These documents satisfy the following EAL2 assurance requirements:

- **ACM_CAP.2**

6.2.2 Delivery and operation

These activities are documented in:

- **SigabaNet 2.2 Installation and Delivery Guide** – Provides complete procedures for securely delivering the TOE to users, including providing evidence of the use of the delivery procedures. Also describes procedures to install and start-up the TOE.

These documents satisfy the following EAL2 assurance requirements:

- **ADO_DEL.1**
- **ADO_IGS.1**

6.2.3 Development

These activities are documented in:

- **SigabaNet 2.2 High-Level Design** – Contains the documentation required to meet ADV_FSP.1, ADV_HLD.1, and ADV_RCR.1. This document provides an informal description of the complete TSF and its external interfaces. This includes the purpose and method of use of all external TSF interfaces and provides details of effects, exceptions and error messages. It also provides a description of the TSF in terms of subsystems. Also describes the security functionality provided by each subsystem and identifies all subsystem interfaces and those that are externally visible. This includes the underlying hardware, firmware, and software required by the TSF and a presentation of the functions provided by the supporting protection mechanisms that they implement. This document also provides the correspondence between the functional specification and the high-level design to demonstrate completeness of each, and to show that the relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.
- **SigabaNet 2.2 Security Policy Model** – Is an informal document that describes the rules and characteristics of all policies of the TOE Security Policy Model that can be modeled. Specifically, it defines the rules between subjects and objects enforced by the discretionary access control policy. These documents satisfy the following EAL2 assurance requirements and the augmentation of EAL2 with an informal security policy model that is required as a dependency of FMT_MSA.2.
- **ADV_FSP.1**

- **ADV_HLD.1**
- **ADV_RCR.1**
- **ADV_SPM.1**

6.2.4 Guidance documents

These activities are documented in:

- **SigabaNet 2.2 Configuration and Administration Guide** – Provides administrator guidance for functions and interfaces available to the TOE administrator. It includes guidance on how to securely administer the TOE, provides warnings about the functions and privileges that should be controlled in a secure processing environment, provides assumptions regarding user behavior that are relevant to secure operation of the TOE, indicates the secure values of security parameters under the control of the administrator, and describes each type of security-relevant event that the administrators need to perform, including changing the security characteristics of entities under the control of the TSF. It also describes security requirements for the IT environment that are relevant to the administrator.

  Also provided in this single document is guidance to users regarding functions and interfaces available to non-administrative users, including the use of user-accessible security functions, warnings about user-accessible functions, and privileges that should be controlled to achieve a secure processing environment. It also describes user responsibilities for secure operation of the TOE and all security requirements of the IT environment that are relevant to the user.

These documents satisfy the following EAL2 assurance requirements:

- **AGD_ADM.1**
- **AGD_USR.1**

6.2.5 Tests

These activities are documented in:

- **SigabaNet 2.2 Test Documentation** – Provides test plans, test procedure descriptions (with tests individually identified and ordered), expected test results and actual test results. Also identifies the security functions to be tested including descriptions of the goal of the tests to be performed. Note that also provided in this single document is the correspondence between the tests and the TSF.

These documents satisfy the following EAL2 assurance requirements:

- **ATE_COV.1**
- **ATE_FUN.1**
- **ATE_IND.2**

6.2.6 Vulnerability assessment

These activities are documented in:

- **SigabaNet 2.2 Vulnerability Analysis**– Provides evidence of a search for obvious vulnerabilities in the TOE, including an explanation of why any identified vulnerabilities cannot be exploited. Also provides an analysis of the strength of the built-in password mechanism.

These documents satisfy the following EAL2 assurance requirements:

- **AVA_SOF.1**
- **AVA_VLA.1**

7 Protection Profile Claims

There are no Protection Profile claims.

8 Rationale

8.1 Security Objectives Rationale

8.1.1 Complete Coverage – Threats

The TOE security objectives have been derived exclusively from statements of organizational security policy, and therefore, there are no explicitly defined threats countered by this security target.

8.1.2 Complete Coverage – Policy

This section demonstrates how there is at least one TOE or IT environment objective for each organizational security policy, and how each objective can be traced back to at least one policy. The following table shows this mapping, and the table is followed by a discussion of the coverage for each policy.

	O.ACCESS	O.ADMIN_ROLE_TOE	O.AUDIT_GENERATION	O.CRYPTO_ALGORITHMS	O.CRYPTO_KEYS	O.DISCRETIONARY_ACCESS	O.MANAGE_TOE	O.USER_AUTHENTICATION	O.USER_IDENTIFICATION	OE.AUDIT_ON_OFF	OE.AUDIT_PROTECTION	OE.AUDIT_REVIEW	OE.TIME	OE.TOE_PROTECTION_IT
P.AUTHORIZED_USERS	X	X		X	X		X	X	X					X
P.NEED_TO_KNOW						X	X							X
P.ACCOUNTABILITY			X						X	X	X	X	X	X

8.1.2.1 P.AUTHORIZED_USERS

Only those users who have been authorized to access the information within the system may access the TOE.

This policy is implemented by the O.ACCESS objective by controlling access to its interfaces using username/password and name assertion authentication mechanisms.

- O.ADMIN_ROLE_TOE objective supports this policy by providing TOE-defined security roles.
- O.MANAGE_TOE objective supports this policy by providing management of TSF data including authentication data.
- O.CRYPTO_ALGORITHMS objectives support this policy by providing a trustworthy means for users to verify their identities, as authenticated by the Authentication Server, to the Key Server.
- O.CRYPTO_KEYS supports this policy by providing cryptographic keys to authorized users to provide secure access to objects in the TOE and the TOE environment.
- O.USER_AUTHENTICATION supports this policy by using its username/password and name assertion authentication mechanisms to verify the claimed identity of administrative and non-

administrative users respectively and by not offering any TSF-mediated functions until the user is authenticated.

- O.USER_IDENTIFICATION supports this policy by providing a unique user identifier for each authenticated user.
- OE. TOE_PROTECTION_IT supports this policy by providing an environment that ensures access attempts must go through the TOE.

8.1.2.2 P.NEED_TO_KNOW

The TOE must limit the access to, modification of, and destruction of the information in protected resources to those authorized users which have a "need to know" for that information.

This policy is implemented by the O.DISCRETIONARY_ACCESS objective, which implements this policy by providing object creator and system administrators with the ability to manage object attributes, and by setting the creator of new objects as owner.

- O.MANAGE_TOE supports this policy by providing administratively-configurable static object security attribute initialization.
- OE. TOE_PROTECTION_IT supports this policy by providing an environment that ensures access attempts must go through the TOE.

8.1.2.3 P.ACCOUNTABILITY

The users of the TOE shall be held accountable for their actions within the system.

This policy is implemented by the O.AUDIT_GENERATION objective, which implements this policy by generating audit records for auditable events.

- O.USER_IDENTIFICATION supports this policy by providing user identifiers to include in audit records.
- OE.AUDIT_PROTECTION supports this policy by protecting the audit trail stored in the environment
- OE.AUDIT_REVIEW supports this policy by providing access to the audit trail stored in the environment.
- OE.TIME supports this policy by providing an environment with a reliable time stamp for audit records.
- OE. TOE_PROTECTION_IT supports this policy by providing an environment that ensures access attempts must go through the TOE.
- OE.AUDIT_OFF_ON supports this policy by ensuring that the operating system generates an audit record when the TOE is started or stopped.

8.1.3 Complete Coverage – Environmental Assumptions

This section provides evidence demonstrating coverage of the Non-IT security objectives by the environmental assumptions. The following table shows this assumption to objective mapping.

	OE.INSTALL	OE.PHYSICAL	OE.CREDEN
A.COOP			X
A.LOCATE		X	
A.PROTECT		X	
A.MANAGE	X		
A.NO_EVIL_ADM	X		
A.PEER	X		
A.CONNECT		X	
A.OS			

8.1.3.1 A.COOP

Authorized users possess the necessary authorization to access at least some of the information managed

The following security objectives for the environment cover this assumption as follows:

- OE.CREDEN supports this assumption by protecting access credentials used for authorization.

8.1.3.2 A.LOCATE

The processing resources of the TOE will be located within controlled access facilities which will prevent unauthorized physical access.

The following security objectives for the environment cover this assumption as follows:

- OE.PHYSICAL supports this assumption by physically protecting the TOE from attack.

8.1.3.3 A.PROTECT

The TOE hardware and software critical to security policy enforcement will be protected from unauthorized physical modification.

The following security objectives for the environment cover this assumption as follows:

- OE.PHYSICAL supports this assumption by physically protecting the TOE from attack.

8.1.3.4 A.MANAGE

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

The following security objectives for the environment cover this assumption as follows:

- OE.INSTALL supports this assumption by initially configuring the TOE such that it is in a state where it can be managed, etc.

8.1.3.5 A.NO_EVIL_ADM

The system administrative personnel are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the administrator documentation.

The following security objectives for the environment cover this assumption as follows:

- OE.INSTALL supports this assumption by ensuring that the TOE is delivered, installed, managed, and operated in a manner that maintains IT security objectives. This precludes the use of administrative personnel who are careless, willfully negligent, or hostile and ensures that the administrators will follow and abide by the instructions in the administrator documentation.

8.1.3.6 A.PEER

Any other systems with which the TOE communicates are assumed to be under the same management control and operate under the same security policy constraints. Conformant TOEs are applicable to networked or distributed environments only if the entire network operates under the same constraints and resides within a single management domain. There are no security requirements which address the need to trust external systems or the communications links to such systems.

The following security objectives for the environment cover this assumption as follows:

- OE.INSTALL supports this assumption by ensuring security is not compromised when installing in a network.

8.1.3.7 A.CONNECT

All connections to peripheral devices reside within the controlled access facilities. Conformant TOEs only address security concerns related to the manipulation of the TOE through its authorized access points. Internal communication paths to access points such as terminals are assumed to be adequately protected.

The following security objectives for the environment cover this assumption as follows:

- OE.PHYSICAL supports this assumption by physically protecting the TOE from attack.

8.2 Security Requirements Rationale

8.2.1 Complete Coverage – Objectives

All Security Functional Requirements (SFR) identified in this Security Target are fully addressed in this section and each SFR is mapped to the objective for which it is intended to satisfy.

8.2.1.1 Security Objectives for the TOE

	O.ACCESS	O.ADMIN_ROLE_TOE	O.AUDIT_GENERATION	O.CRYPTO_ALGORITHMS	O.CRYPTO_KEYS	O.DISCRETIONARY_ACCESS	O.MANAGE_TOE	O.USER_AUTHENTICATION	O.USER_IDENTIFICATION
FAU_EXP.1			X						
FAU_GEN.2			X						
FCS_CKM.1					X				
FCS_CKM.2					X				
FCS_CKM.4				X	X				
FCS_COP.1				X					
FDP_ACC.2	X					X			
FDP_ACF.1	X					X			
FIA_UAU.2								X	
FIA_UID.2									X
FMT_MSA.1							X		
FMT_MSA.2							X		
FMT_MSA.3							X		
FMT_SMF.1							X		
FMT_SMR.1		X							

8.2.1.1.1 O.ACCESS

The TOE will ensure that users gain only authorized access to it and to the resources that it controls.

This TOE security objective is satisfied by ensuring that:

- FDP_ACC.2: All SigabaNet system users are subject to the DAC policy for all available operations on secret keys.
- FDP_ACF.1: Secret keys have owners and ACLs and these attributes are compared against user identities in order to determine whether the request operation should be allowed. If a check fails, access is denied.

8.2.1.1.2 O.ADMIN_ROLE_TOE

The TOE will provide authorized administrator roles to isolate administrative actions.

This TOE security objective is satisfied by ensuring that:

- FMT_SMR.1: Each user account can be assigned zero or more system-defined roles. Any user account that has been assigned one or more system-defined roles is considered a “system administrator” and other user accounts are considered simply “users”.

8.2.1.1.3 O.AUDIT_GENERATION

The TOE will provide the capability to detect and create records of security relevant events associated with users.

This TOE security objective is satisfied by ensuring that:

- FAU_EXP.1: Audit records are generated for the events described in the TSS and audit records include the content described in the TSS.
- FAU_GEN.2: A user identity is associated with each audit event that involves a user.

8.2.1.1.4 O.CRYPTO_ALGORITHMS

The TOE will implement approved cryptographic algorithms for signature generation and verification

This TOE security objective is satisfied by ensuring that:

- FCS_COP.1: The TOE provides pseudo random number generation, signature generation, signature verification, and hash generation.
- FCS_CKM.4: The TOE destroys name assertion signing public and private keys when they are no longer valid, either because they expire or because the administrator deletes them.

8.2.1.1.5 O.CRYPTO_KEYS

The TOE will generate cryptographic keys when requested by an authorized user

This TOE security objective is satisfied by ensuring that:

- FCS_CKM.1: The TOE generates cryptographic keys to serve to requesting user.
- FCS_CKM.2: The TOE distributes message keys to users.
- FCS_CKM.4: The TOE destroys message keys when they are no longer valid.

8.2.1.1.6 O.DISCRETIONARY_ACCESS

The TOE will control access to resources based upon the identity of users.

This TOE security objective is satisfied by ensuring that:

- FDP_ACC.2: All SigabaNet system users are subject to the DAC policy for all available operations on secret keys.
- FDP_ACF.1: Secret keys have owners and ACLs and these attributes are compared against user identities in order to determine whether the request operation should be allowed. If a check fails, access is denied.

8.2.1.1.7 O.MANAGE_TOE

The TOE will provide functions and facilities necessary to support the authorized administrators in their management of the security of the TOE.

This TOE security objective is satisfied by ensuring that:

- FMT_MSA.1: The ability to manage object attributes is restricted to the object creator and system administrators.
- FMT_MSA.2: The ability to accept only secure values for security attributes.
- FMT_MSA.3: By default every object is created with the creator as the owner, and no entries in the ACL. Subsequently, access can be granted to other users by adding entries to object ACLs.
- FMT_SMF.1: The ability to perform the security management functions: create objects, modify an object's ACL, specify alternative initial values for objects, initialize user security attributes, modify user passwords, create name assertions, and revoke object security attributes.

8.2.1.1.8 O.USER_AUTHENTICATION

The TOE will verify the claimed identity of users.

This TOE security objective is satisfied by ensuring that:

- FIA_UAU.2: The TOE offers no TSF-mediated functions until the user is authenticated.

8.2.1.1.9 O.USER_IDENTIFICATION

The TOE will uniquely identify users.

This TOE security objective is satisfied by ensuring that:

- FIA_UID.2: The TOE offers no TSF-mediated functions until the user is identified.

8.2.1.2 Security Objectives for the IT Environment

	OE.AUDIT_ON_OFF	OE.AUDIT_PROTECTION	OE.AUDIT_REVIEW	OE.TIME	OE.TOE_PROTECTION_TOE
FAU_GEN.1	X				
FAU_STG.1		X			
FAU_SAR.1			X		
FPT_RVM.1					X
FPT_SEP.1					X
FPT_STM.1				X	

8.2.1.2.1 OE.AUDIT_ON_OFF

The IT Environment will generate an audit record when TOE auditing is turned off or on.

- FAU_GEN.1: An audit record is generated when the TOE is turned off or on, thereby generating an audit record when TOE auditing is turned off or on. This function is performed by the OS.

8.2.1.2.2 OE.AUDIT_PROTECTION

The IT Environment will provide the capability to protect audit information.

This IT Environment security objective is satisfied by ensuring that:

- FAU_STG.1: The database tables storing the audit trail are protected by the database's own access controls.

8.2.1.2.3 OE.AUDIT_REVIEW

The IT Environment will provide the capability to selectively view audit information.

This IT Environment security objective is satisfied by ensuring that:

- FAU_SAR.1: The database tables storing the audit trail are accessible by authorized administrators who can view them with a utility program or use the SQL select command.

8.2.1.2.4 OE.TIME

The IT environment will provide a time source that provides reliable time stamps.

This IT Environment security objective is satisfied by ensuring that:

- FPT_STM.1: The operating system (kernel) reliably maintains the time.

8.2.1.2.5 OE.TOE_PROTECTION_IT

The IT Environment will protect the TOE and its assets from external interference or tampering.

This IT Environment security objective is satisfied by ensuring that:

- FPT_RVM.1: The operating system and application server ensure that any object access attempts must go through the SigabaNet TOE where the appropriate access rules are enforced.
- FPT_SEP.1: The TOE instantiates itself as a set of servlets running within an application server which protects it from inappropriate access. Incoming client connections are handled using separate connection threads to protect clients from each other.

8.3 Security Assurance Requirements Rationale

The SigabaNet TOE is targeted at a generalized IT environment with good physical access security and competent administrators. Within such environments it is assumed that attackers will have a low attack potential. As such, EAL2 is appropriate to provide the assurance necessary to counter the potential for attack. EAL2 is augmented with FDP_SPM.1 in order to meet dependencies of Security Functional Requirements.

8.4 Strength of Function Rationale

The strength of function rating of SOF-basic is consistent with FIA_UAU.2 by supporting a minimum password length of seven (7) characters, and optionally allowing longer passwords to be used in order to decrease the probability of guessing a password. The Strength of Function Analysis is provided in a separate document to meet assurance requirement AVA_SOF.1

8.5 Requirement Dependency Rationale

The following table represents an analysis of the dependencies of the security functional requirements (SFRs) in this security target. The first column identifies all of the SFRs in this security target. The TOE SFRs are highlighted in bold, unlike the IT environment SFRs. The second column identifies the minimum dependencies defined in the Common Criteria v2.1 and associated interpretations. The third column identifies the actual requirements in this security target that correspond to the identified dependencies. Again, the corresponding TOE SFRs are highlighted in bold. Notice that this table shows that the TOE has some dependencies on the IT environment, but the requirements for the IT environment have been defined such that it is not dependent upon the TOE.

Table 1: Security Functional Requirement Dependencies

Requirement Component	CC Dependencies	ST Dependencies
FAU_EXP.1	None	None
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_STG.1	FAU_GEN.1	FAU_GEN.1
FAU_SAR.1	FAU_GEN.1	FAU_GEN.1
FCS_CKM.1	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4	FCS_CKM.2 FCS_CKM.4 FMT_MSA.2

	FMT_MSA.2	
FCS_CKM.2	[FDP_ITC.1 or FDP_ITC.2 (RI #220) or FCS_CKM.1] FCS_CKM.4 FMT_MSA.2	FCS_CKM.1 FCS_CKM.4 FMT_MSA.2
FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2 (RI #220) or FCS_CKM.1] FMT_MSA.2	FCS_CKM.1 FMT_MSA.2
FCS_COP.1	[FDP_ITC.1 or FDP_ITC.2 (RI #220) or FCS_CKM.1] FCS_CKM.4 FMT_MSA.2	FCS_CKM.1 FCS_CKM.4 FMT_MSA.2
FDP_ACC.2	FDP_ACF.1	FDP_ACF.1
FDP_ACF.1	FDP_ACC.1 FMT_MSA.3	FDP_ACC.2 FMT_MSA.3
FIA_UAU.2	FIA_UID.1	FIA_UID.2
FIA_UID.2	No dependencies	No dependencies
FMT_MSA.1	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	FDP_ACC.2 FMT_SMR.1 FMT_SMF.1
FMT_MSA.2	ADV_SPM.1 [FDP_ACC.1 or FDP_IFC.1] FMT_MSA.1 FMT_SMR.1	ADV_SPM.1 FDP_ACC.1 FMT_MSA.1 FMT_SMR.1
FMT_MSA.3	FMT_MSA.1 FMT_SMR.1	FMT_MSA.1 FMT_SMR.1
FMT_SMF.1	No dependencies	No dependencies
FMT_SMR.1	FIA_UID.1	FIA_UID.2
FPT_RVM.1	No dependencies	No dependencies
FPT_SEP.1	No dependencies	No dependencies
FPT_STM.1	No dependencies	No dependencies

Notes on FCS dependencies:

1. The key generation, import, storage, and destruction dependencies do not apply to the hash generation and random number generation functions of FCS_COP.1, since these functions do not employ keys.

2. Components FCS_CKM.1 Cryptographic key generation, FCS_CKM.2 Cryptographic key distribution, FCS_CKM.4 Cryptographic key destruction, and FCS_COP.1 Cryptographic operation have dependencies on FMT_MSA.2 which are met because the TOE's cryptographic modules are designed to meet FIPS 140-1 standards, which require them to generate and accept only secure security values (i.e., keys). Therefore, the dependency on FMT_MSA.2 is not applicable. The FIPS 140-1 *Security Requirements for Cryptographic Modules* document is available from the National Institute of Standards and Technology website at http://csrc.nist.gov/cryptval/140-1.htm.

8.6 Explicitly Stated Requirements Rationale

The TOE includes one explicitly stated security functional requirement: FAU_EXP.1, Audit data generation explicitly stated. FAU_EXP.1 was created to describe the audit data generation functionality of the TOE. The TOE could not meet FAU_GEN.1, which the CC specifies as the audit data generation SFR, because the TOE does not generate an audit record when the audit function is turned off or on. The TOE auditing function is always on, i.e., there is no way to turn on or off the auditing function, except by turning the TOE on or off. Note that the TOE does generate an audit record when the TOE is turned on (thereby also turning auditing on), but no audit record is generated when the TOE is turned off. FAU_EXP.1 defines the audit data generation functionality of the TOE, since there is audit data generation functionality, but the TOE does not meet all of the functions specified by FAU_GEN.1.

FAU_GEN.1 is included in the IT Environment, since the Operating System, which is in the IT Environment, generates an audit record when the TOE is turned on (thereby turning on TOE auditing) and the TOE is turned off (thereby turning off TOE auditing).

8.7 TOE Summary Specification Rationale

8.7.1 IT Security Functions

Each subsection in Section 6, the TOE Summary Specification, describes a security function of the TOE. Each description is followed with rationale that indicates which requirements are satisfied by aspects of the corresponding security function. The set of security functions work together to satisfy all of the security functions and assurance requirements. Furthermore, all of the security functions are necessary in order for the TSF to provide the required security functionality.

This Section in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions are suitable to meet the TOE security requirements. The collection of security functions work together to provide all of the security requirements. The security functions described in the TOE summary specification are all necessary for the required security functionality in the TSF. The table below demonstrates the relationship between security requirements and security functions. The first column identifies all of the SFRs in this security target. The TOE SFRs are highlighted in bold, unlike the IT environment SFRs.

	FAU	**FCS**	**FDP**	**FIA**	**FMT**	**FPT**
FAU_EXP.1	X					
FAU_GEN.1	X					
FAU_GEN.2	X					
FAU_STG.1	X					
FAU_STM.1	X					
FCS_CKM.1		X				
FCS_CKM.2		X				
FCS_CKM.4		X				
FCS_COP.1		X				
FDP_ACC.2			X			
FDP_ACF.1			X			
FIA_UAU.2				X		
FIA_UID.2				X		
FMT_MSA.1					X	
FMT_MSA.2					X	
FMT_MSA.3					X	
FMT_SMF.1					X	
FMT_SMR.1					X	

FPT_RVM.1						X
FPT_SEP.1						X
FPT_STM.1	X					

8.8 PP Claims Rationale

There are no Protection Profile claims.